Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology - Department of Software

ENHANCING COMMUNITY DETECTION IN TWITTER SOCIAL MEDIA USING TOPIC MODELLING AND SENTIMENT ANALYSIS

A Dissertation
Submitted to the Council of the College of Information Technology - University of Babylon in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Information Technology – Software

By

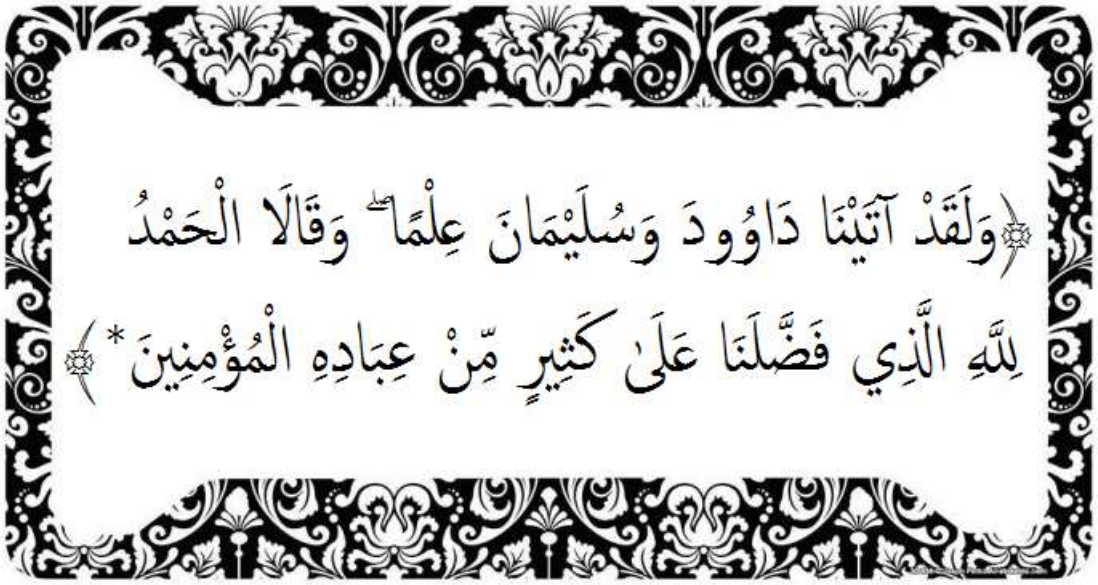
Hayder Majid Abdulhameed Hussain

Supervised by

Prof. Dr. Ghaidaa A. Al-Sultany

2022 A.D. **1443 A.H.**

بسم الله الرحمن الرحيم



صدق الله العلي العظيم

سورة النمل\آية15

Supervisor Certification

I certify that the dissertation entitled “Enhancing Community Detection in Twitter Social Media using Topic Modelling and Sentiment Analysis” was prepared under my supervision at the Department of Software\ College of Information Technology\University of Babylon as partial fulfillment of the requirements of the degree of Doctor of Philosophy in Information Technology - Software.

Signature:
Supervisor Name: Prof. Dr. Ghaidaa A. Al-Sultany
Date: / /2022

The Head of the Department Certification

In view of the available recommendations, I forward the dissertation entitled “Enhancing Community Detection in Twitter Social Media using Topic Modelling and Sentiment Analysis” for debate by the examination committee.

Signature:
Name: Assistant Prof. Dr. Ahmed Saleem Abbas
Head of Software Department
Date: / /2022

Certification of the Examination Committee

We, the undersigned, certify that (Hayder Majid Abdulhameed Alash) candidate for the degree of Doctor of Philosophy in Information Technology - Software, has presented his dissertation of the following title (Enhancing Community Detection in Twitter Social Media using Topic Modelling and Sentiment Analysis) as it appears on the title page and front cover of the dissertation that the said dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on: April 14, 2022.

Signature:
Name: Dr. Saad Talib Hasson
Title: Prof.
Date: / / 2022
(**Chairman**)

Signature:
Name: Dr. Dhia Abdulhussein Alzubaydi
Title: Prof.
Date: / / 2022
(**Member**)

Signature:
Name: Dr. Dheyaa Shaheed Al-Azzawi
Title: Prof.
Date: / / 2022
(**Member**)

Signature:
Name: Dr.Wafaa Mohammed Saeed
Title: Asst. Prof.
Date: / / 2022
(**Member**)

Signature:
Name: Dr. Mahdi Nsaif Jasim
Title: Asst. Prof.
Date: / / 2022
(**Member**)

Signature:
Name: Dr. Ghaidaa A. Al-Sultany
Title: Prof.
Date: / / 2022
(Member and Supervisor)

Signature:
Name: Dr. Hussein Atiyah Lafta
Title: Professor
Date: / / 2022
(**Dean of Collage of Information Technology**)

Declaration

I hereby declare that this Dissertation, submitted to University of Babylon - College of Information Technology -Department of Software in partial fulfillment of requirements for the degree of Doctorate of Philosophy in Information Technology-Software has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose sources are appropriately cited in the references.

Signature:
Name: Hayder Majid Abdulhameed
Date: / / 2022

Dedications

- To My **Mother** and **Father**, for their support, encouragement, and constant love throughout my life. They always trust me and support me.
- To My lovely **Wife**, who leads me through the valley of darkness with light of hope and support.
- To My beloved **Brother** and **Sister**, who stand by me when things look dark.
- To My **Child**, I can't force myself to stop loving you.
- To My **Friends**, who always support me.

Acknowledgements

First of all, I have to express all thanks and gratitude to my God who gives me the ability to achieve such an imperfect work and all what has been done without his blessing and support nothing can be done. I would like to express my sincere gratitude and appreciation to my supervisor **(Prof. Dr. Ghaidaa A. Al-Sultany)** for his invaluable guidance, supervision and untiring efforts during my study.

Special thanks go to all the staff members of Information Technology College for their faithful efforts to give us the utmost scientific topics and endless support in all directions aiming to bring into perfection their scientific followers, and for their unforgettable kind and wise management to our affairs during the research period.

I am immensely grateful to my beloved parents, brother, and sister, who have always prayed for me, and became a guiding force in completing my studies successfully. I want to thank my love and wife, Zainab, for tolerating me in my studies and withstanding the pressure of life and family. I want to thank my wife's family for their encouragement that have supported me in completing my study. I would also like to pay gratitude to the professors who participate as my dissertation examiners committee members, whose suggestions make the research more robust and help significantly improve the dissertation.

Finally, I apologize to those whose names might be missed. But, I am grateful to all of them for the help that I have received from in various research and day-to-day life matters.

Hayder Majid Abdulhameed Alash

Abstract

Nowadays, social media (or social networks) have expanded rapidly. People use social media to share their opinions, ideas, and feelings. They publish various content types like text, images, and videos and interact between them utilizing social attributes such as mentions, hashtags, reposts, and likes. People tend to build clusters based on relationships with others, such as interacting or publishing the same topics or opinions. Community detection was presented as a method to find communities in social networks by dividing a network into groups of closely connected users. Community detection is a big challenge when analyzing social media due to its structure and diversity of resources. The current studies concentrate on a structural network which only depends on links between users to detect communities. These works may not analyse the communities well because they depend only on links and nodes in the network.

The proposed model was developed and exploited different techniques to enhance the Leiden community detection method. The first technique exploits the Latent Semantic Analysis algorithm (LSA) with hashtags to discover users topics to improve community detection performance. Using LSA with community detection gives the communities more contextual information. Second, employing mention and retweet attributes to build interaction networks. Third, a Support Vector Machine (SVM) is utilized to get users opinions about a particular topic. Fourth, the topic, sentiment, and retweet with mention were exploited to calculate

a new weighted network. Finally, a new weighted network can enhance the quality function of modularity and a Constant Potts Model (CPM) for community detection methods.

Several experiments have been done on different Twitter and benchmark networks. The results have shown improvement in the quality of community detection for modularity and CPM when the topics, sentiment, retweet, and mention exploit to get a new weighted network. A new weights network calculation increases the quality of a Leiden method for modularity from 0.7984 in link relationship to 0.8234 and CPM from 0.7514 and 0.8488 in link relationship to 0.9133 and 0.9637. whereas, A new weights network calculation increases the quality of a Leiden method on the Zakary Karate Club dataset for modularity from 0.4198 in link relationship to 0.4647 and CPM from 0.5282 and 0.6865 in link relationship to 0.7447 and 0.8295.

Table of Contents

Dedications iii
Abstract v
Table of Contents vii
List of Tables x
List of Figures xi
List of Algorithms xii
List of Abbreviations xiii
List of Symbols xiv
List of Publication xv
Part of Speech Symbols xvi
Chapter One Introduction 1
1.1 Overview 2
1.2 Motivation 5
1.3 Challenges 6
1.4 The Aim and Objective of Dissertation 7
1.5 The Contributions of the Dissertation 7
1.6 Related Works 8
1.7 Dissertation Organization 19
Chapter Two Social Networks Analysis and Community Detection 20
2.1 Overview 21
2.2 Social Networking 21
2.3 Content Analysis of Social Network 25
2.3.1 Preprocessing 28
2.3.2 Term Frequency - Inverse Document Frequency (TF-IDF) 31
2.3.3 Latent Semantic Analysis (LSA) 32
2.3.4 Support Vector Machine (SVM) 35
2.4 Structural Analysis of Social Network 37
2.4.1 Louvain Method 41

2.4.2 Leiden Method 45
2.5 Hybrid Analysis of Social Network 50
2.6 Evaluation Metrics of Social Network Analysis 52
Chapter Three Design and Develop Community Detection Model 57
3.1 Overview 58
3.2 Community Detection Model 58
3.3 Twitter Collection 61
3.4 Twitter Data Preprocessing 62
3.5 Twitter Network Construction 65
3.6 Topic Model 67
3.6.1 Hashtags Enrichment 68
3.6.2 TF-IDF Measurement 68
3.6.3 Latent Semantic Analysis (LSA) 69
3.6.4 Evaluation of LSA Model 70
3.7 Sentiment Analysis 72
3.7.1 Support Vector Machine (SVM) 72
3.7.2 Evaluation of SVM 73
3.8 Community Detection 74
3.8.1 Enriching Community Network 75
3.8.2 Enhancing Leiden Community Detection 79
3.8.3 Evaluation of Community Detection 81
Chapter Four Experimental Results and Discussions 82
4.1 Overview 83
4.2 System Requirements 83
4.3 Twitter Datasets 83
4.3.1 Data Extraction 85
4.4 Use Case Study 87
4.4.1 Preprocessing Stage 87
4.4.2 User Network Formation 91

4.4.3 User Topic Discovery 94
4.4.4 User Opinion Classification 99
4.4.5 Enhancing Community Detection 102
4.5 Other Experiments 108
4.6 Benchmark Dataset 113
4.7 Results Comparison 115
4.7.1 LSA and LDA Topic Modelling 115
4.7.2 Leiden and Louvain Community Detection Algorithms 117
Chapter Five Conclusion and Future works 119
5.1 Conclusion 120
5.2 Future Works 121
REFERENCES 124
الخلاصة 134

List of Tables

Table 1. 1. Summary of Related Work 17
Table 4. 1. Details of Raw Data Collected 84
Table 4. 2 Sample of CSV Tweets 86
Table 4. 3. Sample of Preprocessing Operation 88
Table 4. 4. Example of Tweets in Dataset 89
Table 4. 5. Result of Preprocessing on Tweets 90
Table 4. 6. Result of POS-Tagging on Tweets 90
Table 4. 7. Retweets' Weight in the Network 92
Table 4. 8. Mention s' Weight in the Network 93
Table 4. 9. LSA Coherence Value Comparison 95
Table 4. 10. LSA Topics Result with Enriched Tweets 97
Table 4. 11. Dominant Topics on Tweets 98
Table 4. 12. Labeled Tweets 100
Table 4. 13. Opinion of Users Tweets 101
Table 4. 14. Leiden Community Detection Result of Dataset 1 105
Table 4. 15. Coherence Score for LSA on Dataset 2 110
Table 4. 16. LSA topics result on Enriched Tweets 111
Table 4. 17. Community Detection Result on Dataset2 113
Table 4. 18. Zachary Network Result 114
Table 4. 19. Comparison between LSA and LDA Results 116
Table 4. 20. Comparison Result of Community Detection 118

List of Figures

Figure 1.1 Communities in Social Network 3
Figure 2. 1 Twitter Tweet and Features [41] 23
Figure 2. 2 Decomposition Word-Document Matrix [10] 33
Figure 2. 3 LSA Steps [10] 35
Figure 2. 4 SVM Method [64] 37
Figure 2. 5 Louvain Method [17] 45
Figure 2. 6 Louvain Arbitrarily Poorly Connected [17] 46
Figure 2. 7 Leiden Method [17] 50
Figure 3.1 Methods for Improve Social Network Analysis 59
Figure 3. 2 Proposed Model 60
Figure 3. 3 Sample Graph Network 67
Figure 3. 4 New Weight Network 78
Figure 4. 1 Simple JSON Format Tweets 85
Figure 4. 2 Sample Retweet Network 92
Figure 4. 3 Sample Mention Network 93
Figure 4. 4 LSA Topic Model Process 94
Figure 4. 5 Coherence Score of LSA for Different K 96
Figure 4. 6 LSA Word Cloud for Topics on Enriched Tweets 98
Figure 4. 7 SVM Sentiment Result 101
Figure 4. 8 Enhanced Detecting communities 104
Figure 4. 9 Community Detection Result Details 106
Figure 4. 10 Distribution of Sentiment in Communities 108
Figure 4. 11 Coherence Score of LSA for Different K 109
Figure 4. 12 LSA Word Cloud for Topics on Dataset 2 111
Figure 4. 13 SVM Sentiment Result on Dataset 2 112
Figure 4. 14 Zachary Communities 114
Figure 4. 15 LSA and LDA coherence Score Results 117

List of Algorithms

Algorithm 2. 1 Louvain Algorithm 44
Algorithm 2. 2 Leiden Algorithm 48
Algorithm 3. 1 Collect Data from Twitter Streaming API 62
Algorithm 3. 2 Preprocessing 63
Algorithm 3. 3 Build Retweet Network, Mention Network, Hashtags 66
Algorithm 3.4 LSA Topic Model 70
Algorithm 3.5 Topics Coherence 71
Algorithm 3.6 SVM Method 73
Algorithm 3.7 New Weighted Network Calculation 77
Algorithm 3.8 Enhancing Leiden Community Detection 80

List of Abbreviations

ABBREVIATION	MEANING
API	Application Program Interface
ASN	Algorithm Similarity Network
BOW	Bag of Words
CPM	Constant Potts Model
CSV	Comma Separated Values
DMM	Dirichlet Multinomial Mixture
FGM	Fast-Greedy optimization of Modularity
JSON	JavaScript Object Notation
LDA	Latent Dirichlet Allocation
LPA	Label Propagation Algorithm
LSA	Latent Semantic Analysis
NB	Naive Bayes
NMF	Nonnegative Matrix Factorization
PLSA	Probabilistic Latent Semantic Analysis
POS	Part-of-Speech
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TF-IDF	Term Frequency - Inverse Document Frequency

List of Symbols

SYMBOL	DESCRIPTION
#	Hashtag
@	Mention
RT	Retweet
K	Number of topics
T	Topic
U	User
V	Vertices
E	Edge
W	Weight

List of Publication

1. H. M. Alash and G. A. Al-Sultany, “Improve Topic Modeling Algorithms Based on Twitter Hashtags,” in *Journal of Physics: Conference Series, 2020*, vol. 1660, no. 1, p. 12100.
2. H. M. Alash and G. A. Al-Sultany, "Enhanced Twitter Community Detection using Node Content and Attributes," *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*, 2021, pp. 5-10, doi: 10.1109/BICITS51482.2021.9509873.

Part of Speech Symbols

SYMBOL	DESCRIPTION
CC	Coordinating Conjunction
CD	Cardinal Digit
DT	Determiner
EX	Existential
FW	Foreign Word
IN	Preposition/Subordinating Conjunction
JJ	Adjective
JJR	Adjective, Comparative
JJS	Adjective, Superlative
NNP	Proper Noun, Singular
NNPS	Proper Noun, Plural
PDT	Pre Determiner
POS	Possessive ending
PRP	Personal Pronoun
PRP$	Possessive Pronoun
RB	Adverb
RBR	Adverb, Comparative
RBS	Adverb, Superlative
UH	Interjection
VB	Verb, Base form
VBD	Verb, Past tense
VBG	Verb, Gerund/Present Participle
VBN	Verb, Past
WP	Wh-Pronoun
WP$	Possessive wh-Pronoun
WRB	Wh-adverb

Chapter One

Introduction

1.1 Overview

In the recent decade, social networks have grown very rapidly. Millions of people are interested in using social networks such as Twitter and Facebook. The structure of social networks is interesting because it provides insight into how users interact with others. Social networks let people connect over the internet to share their opinions, feelings, and ideas. Users can easily share information, connect with others, and participate in online discussions. Users tend to create groups to exchange thoughts or events on social networks [1]. Awareness of these social subgroups provides additional insight into the structure of these networks. It's similar to the idiom, "Birds of a feather flock together," that people with similar interests, personalities, characters, or other distinguishing characteristics tend to associate with one another, as seen in Figure (1.1). Each color represents a community of users connected closely between them and less with other users in other communities.

The Twitter social network was established in 2006 with a few thousand users, but recently, it includes over 190 million active users on Twitter daily [2]. Twitter is short text (post), including misspelling words, abbreviations, emoticons, and non-conventional syntax. Replies, mentions, hashtags, and retweets examples of features on the Twitter social network. Twitter data has grown over the years, but all this data is worthless unless analyzing what users post and react on Twitter by finding topics, sentiment, or clusters (communities). The extraction of critical information and connections from Twitter datasets is still a big challenge.

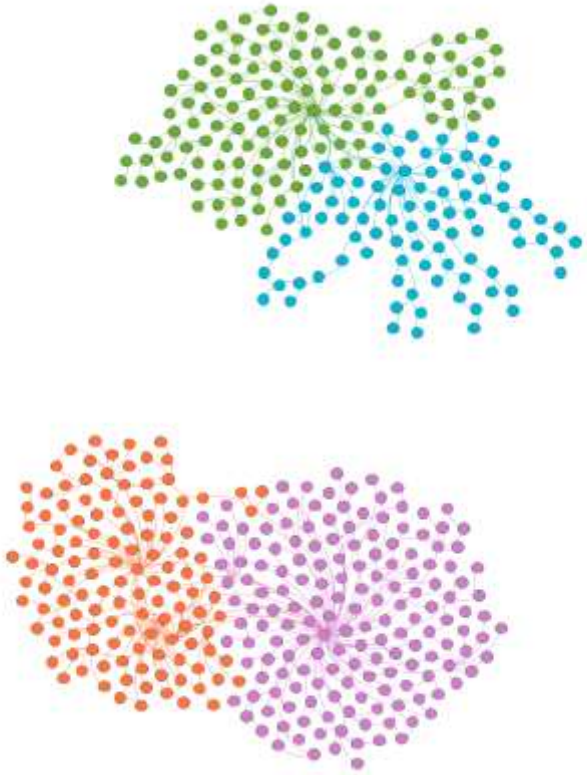


Figure 1.1 Communities in Social Network

Therefore, understanding the structural and interaction of Twitter social networks is critical in research. A Twitter social network consists of users represented by vertices that link and interact with other users or organizations by edges. A community represents a graph in which vertices represent a set of users and edges represent the relations between users [3]. Users create communities in a social network. Communities are groups of vertices connected more closely between them than other vertices in the network [4]. Community detection is a process of grouping similar users into the same cluster. However, this process is a difficult task in the social network. Several communities detections methods were

proposed starting from Newman [5], the Louvain algorithm [6], Label Propagation [7], and Infomap [8]. These approaches have used only the link structure of the network. They may not accurately determine the community members because of the sparse and noisy network topology of social networks such as Twitter. The network changes rapidly due to dynamic changes in users interaction.

Other researchers have used only the user content to determine topics (communities) in large documents such as Latent Dirichlet Allocation (LDA) [9], Latent Semantic Analysis (LSA) [10], Probabilistic Latent Semantic Analysis (PLSA) [11], and Nonnegative Matrix Factorization (NMF) [12], but the results are inappropriate because these topics model algorithms only depended on the text [13]. Therefore, deriving topics from the Twitter short text (tweets) is a big challenge. The number of words to build frequency of co-occurrences between terms is low. Furthermore, some researchers used text content to classify user opinion into positive, neutral, and negative. Users frequently share their sentiments about specific events and topics on the social network. Sentiment analysis algorithms such as Naive Bayes (NB) and Support Vector Machines (SVM) were used to find users sentiments [14].

These algorithms may not analyze and determine the communities well because they only depend on one factor, such as users' text or link relationships among users. The social networks have many user contents and attributes that should not be taken as separate from the structure of network during analysis. Some researchers addressed the problem of lack of content by considering user content to supplement link networks in

community detection by combining them to improve the quality of community detection algorithms [15][16].

However, community detection algorithms are an essential process in social networks used to detect user communities. In this dissertation, the work has developed a model for combining user content and user attributes to enhance the performance of social network communities detection.

1.2 Motivation

One of the most critical questions in social networks is communities. How can determine clusters of the network that users are connected closely between themselves and less connected with the rest of the network? How users' properties and interaction with each other effect on detection and performance of community detection algorithms.

The vast and rich data on social networks call for developing communities detection algorithms and analyzing user content and attributes by looking at how network structure is affected by user content and user attributes. Communities detection algorithms have shortcomings when analyzing social networks due to only depending on link relationships among users. Users have the text and other information (mention, replay, hashtag) that provide more data to consolidate the connection on social networks. This dissertation attempted to utilize content and attributes on the Twitter social network by combining topic modelling and sentiment analysis with link relation to enhance community detection.

1.3 Challenges

Recently, community detection has been one of the interesting areas of networks analysis because of the rapid growth of the social networks. There are many challenges in this work:

1- There is a lack of social networks datasets that have users' content and relationships in the same dataset. Furthermore, The ground truth of structure and text dataset is nonexistent.

2- Collecting data from social networks faced some difficulty. Several posts contain no text, such as pictures or emoji. Some posts were advertisements of events or company products. Many posts are duplicated text. Therefore, social networks data needs several preprocessing to convert unstructured data into structured data acceptable for analysis.

3- The labeled data is scarce, and unlabeled information is abundantly available. Therefore, data should be manually classified to train supervised classification algorithms to find the opinion of users.

4- Incorporating content and other attributes on community detection algorithms is complex and need multiple data resources.

The biggest problem with discovering communities is when it only depends on structural networks. It does not help study and analyze the community more accurately and know how to market through it. Thus, it leads to a lack of information that can help know the discussion topics and users' feelings within the communities.

1.4 The Aim and Objective of Dissertation

The detection of communities intends to find a group of closely connected users in the network published on Twitter. Communities detection assists analysis of users' information on Twitter. Enhancing and developing the community detection algorithms strategy was the main aim of the work in this dissertation.

This work's objective is to create communities in a new way based on contextual content and structural network. This information helps to analyze and enhance the discovery of communities using the community detection algorithm that allows companies and researchers to analyze customers' opinions about specific products or topics rather than only depending on network relationships.

Therefore, the proposed model uses content and attributes to improve communities' detection using the topic modelling technique and sentiment analysis with retweets and mention Twitter attributes. Topic model and sentiment provide more information that can help update the edges weights of the network. The current model utilizes a new weighted network to enhance the quality of modularity and a Constant Potts Model (CPM) for Leiden community detection.

1.5 The Contributions of the Dissertation

This study will help researchers and companies enhance and develop community detection algorithms by adding more helpful information on communities rather than only dependent on network relationships. The contribution of this dissertation are summarized as follows:

1- Improving community detection algorithms for identifying communities in a new way based on contextual content and structural network by incorporating users' topics and opinion information with retweets and mention networks to calculate a new weighted network which has helpful information.

2- Detection has been improved through the LSA that exploits hashtags to discover hidden topics in the tweets.

3- Using sentiment analysis to reveal the users' trends in what is appropriate for the researcher's work in these communities. Some trends were set manually by tagging tweets to create a training dataset that the sentiment methods use.

4- Construct dataset by downloading tweets from Twitter and processing them to build valuable data in this work. This dataset contains texts, relationships, and trends that can use by other researchers in future works.

1.6 Related Works

Most previous social network research has focused on discovering the communities from the structure networks using community detection algorithms [3][17]. The traditional community detection algorithms have been successful in identifying communities from social networks such as Newman [5], Louvain algorithm [6], Label Propagation [7], Infomap [8], and Fast Greedy [18]. However, these approaches only looked at one feature (links) while neglecting other features to detect communities in the social network. Further research proposed a solution to this problem

by integrating users' content information with link relationships in communities detection algorithms to increase the quality of communities detection. All the related studies are listed below with additional details and summarized in Table (1.1).

1- The authors in [3] suggested a Balanced Link Density-based Label Propagation algorithm (BLDLP) developed from the Label Propagation Algorithm (LPA). It used a balance parameter that replaced the random selection of labels. This method addressed the uncertainty problem of random sampling in the LPA using different values of a balance parameter to achieve various quality outputs. The proposed approach benefits from certainty and stability in the output results. In contrast, its time complexity is still close to the fundamental LPA and less than other methods.

2- The authors in [17] built a new Leiden algorithm that overcome the community connects badly problem on the Louvain algorithm. Node communities detected by the Louvain algorithm may disconnect from each other. Leiden is an iterative algorithm that guarantees that nodes inside communities are well connected and that all subsets communities are assigned optimally. The algorithm exploits the notion of fast local moving and moving nodes to neighbors randomly to reduce time complexity and improve detect quality. The results showed that the Leiden method is faster than the Louvain method and detects more suitable communities.

3- In [19], The authors proposed three stages method that proposed combining the central nodes identification, the label propagation, and

community merger stages. Central node identification identifies central nodes based on the distance between nodes. The label propagation represents nodes labeled with the same colors if they accomplish maximum similarity. The two communities merge if they increase the modularity. The method was tested on the real world and synthetic networks. The results showed that three stage model performs better than other widely used methods. This approach has high complexity for large networks.

4- In [20], the authors studied online user discussion on Twitter to detect the user community's opinions about vaccination. This work analyzed Twitter data first, then used the retweet network, which represents the user interactions. For this aim, two aspects are utilized: topic relevance factor and kurtosis of vaccination coverages. Topic relevance factor determines the importance of vaccine topics in a specific country. Kurtosis is estimating the diffusion changes of vaccination coverages rates. Community detection algorithms are applied on retweet networks to identify vaccine communities. Various network metrics are authors calculated to identify the most relevant users and analyze their social effects. The results showed that these approaches could be utilized to find social conversation communities providing helpful knowledge to enhance immunization techniques.

5- The authors in [21] proposed Attracting and Recommending Degree (AR-Cluster) graph clustering algorithm to detect communities in social networks. A novel collaborative similarity metric is used to cluster nodes together in the AR-cluster approach based on calculated

similarities among vertices to calculate node similarities. A K-Medoids framework adopts to clustering graph. The results have shown that the AR-Cluster approach performs well compared to the other three techniques (Weights (W) Cluster, Similarity Attributes (SA) Cluster, Local- Global cluster- Collaborative Similarity Measure (LGC-CSM)). This approach is challenging to use for large social networks.

6- In [22], the author presented classifying community discovery methods based on the similarity of their results. They built Algorithm Similarity Network (ASN), in which vertices are the community detection algorithms and edges set between them if they return similar groupings. This work attempts to build a similarity based categorization of community detection techniques that two methods provide identical communities. The results showed that the ASN discovered well-separated groups compared to other methods.

7- The authors in [23] provided a methodology for analyzing the performance of community discovery methods using visual analysis and statistical to help decision making. Infomap and Louvain, two community detection algorithms, are tested on four real-world networks datasets with various characteristics to identify appropriate community detection algorithms for a specific dataset. The results showed the similarities and differences between algorithms using statistical and visual analysis to help decision-making.

8- Local and Global node Influence-based community detection (LGIEM) proposed in [24] to discover communities. A centrality

measure based on local and global knowledge LGI is used to discover influential users by local attributes and global attributes. Overlapping communities merged to obtain the cluster construction. This method evaluated the node influence method (LGI) using Susceptible–Infected–Removed (SIR) distribution model. The results showed a new technique, LGIEM, more accurate than other similar techniques. This approach is time complexity for large social networks.

9- The authors in [25] proposed a new overlapping community detection algorithm based on density peaks (OCDDP) to detect communities. A distance matrix computation approach is initially presented. After that, a three-step technique for selecting community cores is used to calculate the centers of the clusters. Finally, they developed a node allocation mechanism based on membership vectors. The efficiency of OCDDP performs well when dealing with simple networks compared with the existing techniques, while OCDDP still performs well with complex networks.

10- In [26], the authors proposed a novel framework for Twitter user clustering to discover political preferences by combining community detection with the force-directed graph method to find political clusters. Hard links and soft links are used to construct networks. A hard link is a link between two users who follow each other. A soft link is a link between two users if one user follows the other. The quality of this clustering technique is assessed against a subset of human-labeled user profiles.

11- In [27], a hierarchical clustering procedure is built on user relationships and interests networks to find the communities. They used semantic analysis and the following/follower relationships to calculate edge weight to improve community detection. The original network changed into a new network, including the undirected and weighted edges. The weights generate using the direction and interest vectors in the original network, and the edge weights are used to find the similarity between edges. The hierarchical clustering algorithm is used to identify communities based on the edge-weighted similarity. The results showed that the proposed method performs better than the CF method, improving the precision.

12- A new technique was proposed in [16] to identify influential communities in social networks by using emotional behavior and users' profile. This method depended on the content of each user post to identify user emotions based on the influence metric of the user on a particular topic. This approach assigned each user a specific influence metric to detect the most influential communities. The method outperforms the popular modularity optimization method used in related studies. This technique suffers when analyzing a complex social network.

13- The authors in [28] studied the impact of sentiment analysis on users' behavior in social networks. They used three Twitter datasets related to people who share links between them and people who talked about a specific subject. Sentiments have analyzed the relation between people to positive, negative, or neutral. Then, a k-core community

detection algorithm is used to discover communities, then track changes in users' sentiments in communities. In all three datasets, the result has shown a strong link between positive sentiment and community size. The positive sentiment is spreading more frequently on communities members than negative sentiment over time. Its effect on the community performance includes birth, merge, split,…etc.

14- In [29], the authors have proposed a novel method called Multi-View Clustering via Robust Nonnegative Matrix Factorization (MVCRNMF) that combines content information and link to discover community. This method used the link and content in users' data to complete each other and form a multi-view robust Non-negative Matrix Factorization (NMF) model with co-regularized constraints. This technique can understand the contribution weights from content and link data, saving much work on adjusting the weight value. The results showed that MVCRNMF achieves better than state-of-the-art approaches and gets higher-quality communities. This approach is time complexity for large social networks.

15- The authors in [30] enhanced the community detection algorithm by integrating a semantic similarity with Fast-Greedy optimization of Modularity (FGM) to deal with the complex social network. The enhanced FGM has combined with the spatial clustering of applications to discover geo-located communities within the detected topical communities. This approach has limited analysis of large networks because of the manual clustering of text. The results showed that communities relevant to specifying locations where reported

disaster-related happenings could be extracted. The enhanced algorithm is better than the generic one in this task.

16- The authors in [31] showed how to detect specific topics by applying the social network detection method. Communities contain different interests and content that use flow-based community detection techniques to identify groups in the debate concerning health care. The markov stability and role-based similarity methods are applied to analyze a set of Twitter care datasets that offer information to policymakers on how to connect successfully with a Twitter audience. The results showed Twitter is effectively a communication medium.

17- The authors in [32] enhanced community detection by combing knowledge graphs. They combined metrics of the context information of users with other attributes using hierarchical concept maps, which are used to influence the search for perfect concept generalization. The similarity node attributes are integrated with a variety of generalized concepts for detecting communities. A community detection technique improves topic generalization and community structures. The results showed that the proposed method improves the F-measure and Jaccard 20% better than the existing state-of-the-art of community detection techniques.

18- The authors in [33] proposed a new method to detect local communities and common interests of users using a constructed graph by an edge attributed multilayer network of Twitter membership lists. This approach found communities semantically homogeneous and identified targeted and overlapping communities from a local

multilayer network. The results showed that the proposed method performs better than global community detection approaches. Also, It is fast as good as local community detection approaches. This method neglected the temporal development of local communities.

19- The authors in [34] presented a new approach that considers user interactions and message content to detect communities. This method represented interactions with users by retweet relations and messages contents with semantic similarity of users. They used topics exchanges between users to improve detects communities using traditional community detection methods.

20- The authors in [35] proposed applying topic model and community detection in Twitter to cluster similar users who discuss the same topics and communities from the Human Papilloma Virus (HPV) injections network. Dirichlet Multinomial Mixture (DMM) and latent Dirichlet allocation methods with Louvain and Infomap community detection used to discover communities and topics of the users from social networks. This approach provides a helpful way to describe the community's opinion in public health applications. The results showed that the proposed model achieved higher alignment values while the number of topics was lower. The limitation of this approach did not consider the temporal topics or the community structure in edges weights.

Table 1. 1. Summary of Related Work

Authors	Techniques	Community Detection Alg.	Datasets	Evaluation
E. Jokar and M. Mosleh, 2019 [3]	Density links feature	BLDLP	Zachary's Karate Dolphins American Football Facebook network	Modularity Normalized Mutual Information (NMI)
V. A. Traag, et al., 2019 [17]	Fast local move approach	Leiden	DBLP Live Journal Amazon Web UK Web of Science IMDB	Modularity CPM
X. You, Y. Ma, and Z. Liu, 2020 [19]	Integrating label propagation, central nodes identification, and the cluster uniting steps	Fast greedy, Infomap, Eigen vector, LPA, Walk trap, Louvain Node2vec TS algorithm	LEF synthetic networks Zachary's Karate Dolphin Polbooks American Football	Modularity Normalized Mutual Information (NMI)
G. Bello-Orgaz, et al., 2017 [20]	Kurtosis of Vaccination Coverage Rates, Topic Relevance Factor	Fast Greedy Label Propagation Multi-Level Walktrap InfoMap Leading Eigenvector	Twitter WHO Web Site	Modularity Density Cohesion Omega
H. Zhou, et al., 2017 [21]	Structural and attribute aspects a random walk strategy.	Attracting and Recommending Degree (AR-Cluster)	Political Blogs DBLP	Density Entropy NMI
M. Coscia, 2019 [22]	Similarity Network	InfoMap, Edge betweenness , Walktrap	LEF synthetic networks	Modularity NMI
C. D. G. Linhares, et al., 2020 [23]	Statistical and visual analysis to help decision-making	Infomap and Louvain	primary school in Lyon high school in Marseille	Precision Recall F-Measure Modularity
T. Ma, Q. Liu, et al., 2020 [24]	Centrality measure, seed expansion of a community discovery method	(LGIEM) Louvain LPA	LEF synthetic Zachary's Karate Dolphin Polbooks American Football	F-Measure Modularity NMI Accuracy
X. Bai, P. Yang, and X. Shi, 2017 [25]	Density peak clustering method	overlapping community detection method (OCDDP)	LEF synthetic Zachary's Karate Dolphin Polbooks Netscience Email	NMI

D. L. Sánchez, et al., 2016 [26]	hard links or soft links	Louvain	Twitter	Purity
C. Li, et al., 2019 [27]	user relationships and interests networks	Hierarchical clustering based on edge-weighted similarity	Weibo	Precision Recall F-Measure
A. Kanavos, et at., 2018 [16]	emotional behavior and users' profile to calculate influence metric	Louvain	Twitter	Modularity
N. Alduaiji and A. Datta, 2018 [28]	Common interest of users with relationship	K-core	Twitter	Correlation
Y. Tang, et al., 2019 [29]	Combines content information and link	Multi-View Clustering via Robust Nonnegative Matrix Factorization (MVCRNMF)	CiteSeer Cora WebKB	Density Entropy
M. Bakillah, et al., 2015 [30]	Semantic similarity	Fast Greedy optimization of Modularity (FGM)	Twitter	Modularity Recall Purity
B. R. C. Amor, et at., 2016 [31]	Markov Stability and Role-based similarity methods with common interest and relationships	Flow-based community detection	Twitter	Degree
S. Bhatt et al, 2019 [32]	Incorporate context information of users with other attributes using hierarchical concept maps	Louvain UNCut CPCD JCDC	G+ ego network Twitter DBLP Reddit	F-Measure Jaccard measure
Benabdelkrim, et at., 2020 [33]	Constructed graph by an edge attributed multilayer network	Louvain LPV	Twitter	Modularity
Thi Hoang, 2020 [34]	Embedding users and messages content	InfoMap, Label Propagation, Leading Eigenvector, Louvain, Spin glass, Walktrap	Twitter	Modularity NMI F-divergence
D. Surian, et al., 2016 [35]	LDA, Dirichlet Multinomial Mixture (DMM)	Louvain Infomap	Twitter	NMI Purity

1.7 Dissertation Organization

There are five chapters in the dissertation. Each chapter begins with a brief introductory overview of the subject. The remaining parts of each chapter are as follows:

- **Chapter 2:** covers general information on social network analysis, topic modelling, sentiment analysis, and community detection techniques. The tools used for evaluating these techniques in social networks.
- **Chapter 3:** Illustrates the proposed improved community detection algorithm model by integrated topic modelling, sentiment analysis, and link relationships of users on Twitter.
- **Chapter 4:** presents and discusses the experimental results of the enhancing Leiden community detection model on a Twitter social network.
- **Chapter 5:** describes the dissertation-derived conclusions and provide recommendations for future work.

Chapter Two

Social Networks Analysis and Community Detection

2.1 Overview

Many social networks websites have emerged that allow communication among users and create communities discussing different aspects of users' lives. These communities are used to exchange ideas, thoughts, collaborate, and make friends via social networks. Users can share and discuss a particular event or concept, such as a new movie in a cinema, a new product, a specific accident, or anything in the real world by posting text, images, and videos. The development of social networks sites has led to appearance need to analyze what people talk about and interact between them by finding topics under discussion and discovering communities. Community detection analysis is the procedure of identifying the clusters or communities on social networks. communities is one of the most crucial components of a social network. Graphs are used to analyze relationships between users on social networks to cluster users in different communities, which can be helpful in many applications in the real world. This chapter provides background information about the analysis of the online social network. It focuses on models and algorithms used to analyze social networks. The topic model derives topics from short texts such as LSA. Then, sentiment analysis methods classified the feeling of users about topics into texts and the community detection algorithms used to detect communities.

2.2 Social Networking

Today, researchers consider social networks a necessary component of connections between people in the world. The network formation and spread of information have grown research rapidly that studies online

social networks. The primary step for social network analysis is the large-scale analysis of communication links, text contents, and information spreads. There are millions of active daily people on social networks. Social networks are social structures of people or companies linked by relationships representing users' shared interests or ideas [36]. It consists of users defined by nodes (vertices) that link and interact with other users or organizations by the edge [37]. The structure of these networks is interesting because it provides insight into how people interact with one another. People tend to have a lot of mutual friends, which creates a social network. People with similar interests, opinions, and choices are more likely to connect in a social network, making different virtual communities or clusters. Communities are a set of more closely connected vertices than the rest of the network's vertices. The technique of grouping similar users into a cluster is known as community detection.

Many people in social networks like Facebook and Twitter own billions of interactions among them, making analyzing these communications more complicated. Twitter has emerged as the most popular platform for research among social networks because most data on Twitter is public by default and can be quickly accessed via the Twitter Application Program Interface (API) [38]. Twitter is one of the active social networks regarded as one of the top-rank online social networks than others. It is a microblogging website that allows users to share their opinions and ideas known as "tweets" [39]. A tweet is a short text restricted to 140 characters, but the limitation was doubled to 280 characters in 2017 for non-Chinese, Japanese, and Korean languages [40]. Unfriend users can just read publicly observable tweets, while friend users can post, reply, comment,

and retweet text. Any user can be followed by another user. As a result, each user has a group of followers (people who get your tweets) and followings (people whose tweets show on your page). Tweets can contain images, URLs, and videos. In addition, it may include several other features such as hashtags, retweets (RT), mentions, and replies (see Figure 2.1).

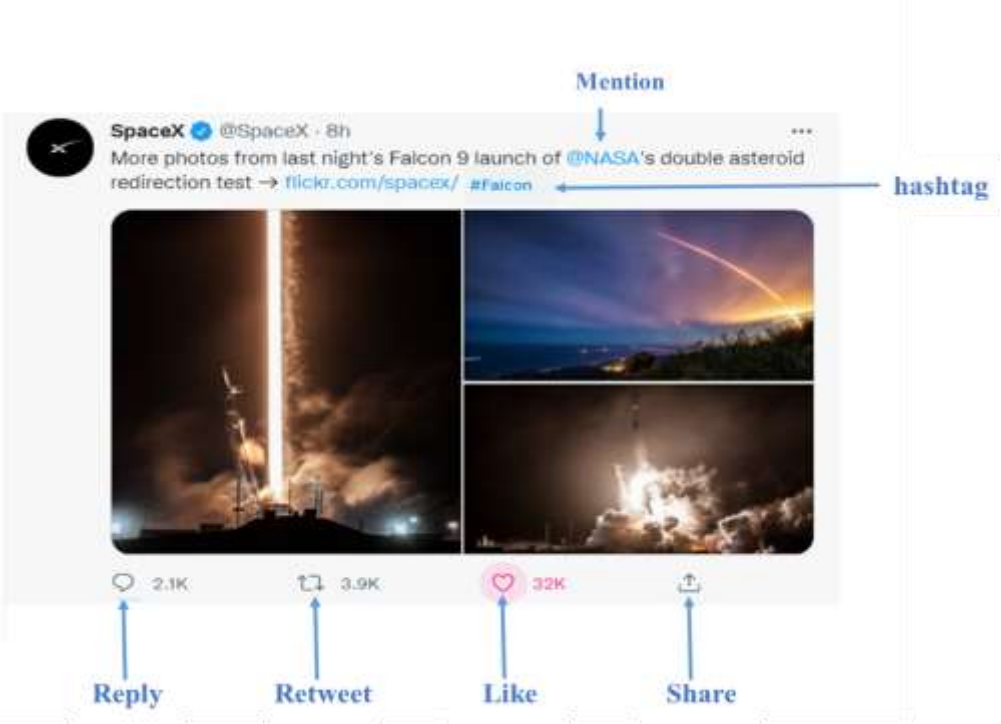


Figure 2. 1 Twitter Tweet and Features [41]

A hashtag is a "#" symbol followed by a relevant keyword or phrase (i.e. '#nature'). Hashtag highlighted the significant content in tweets, allowing users to classify them and find them more quickly in search. Hashtags can indicate topics, concerns, and issues that the user has emphasized. Hashtags have become a social concept. Several social networks use them as a simple way to express an idea, opinion, event and

represent term (or phrase) in a brief text. The process of posting hashtags to individuals, locations, or events refers to social tagging. Therefore, Twitter allows users to see all tweets that include a particular hashtag by creating a group of tweets with that hashtag. Users can use more than hashtags in tweet posts to indicate the topic of the published message.

A Retweet (RT) is re-post another user tweet. The retweet feature allows users to fast share the tweet with all users who follow the tweet's author. The simplest and formal way to retweet a post is to click the retweet button at the tweet. In an informal approach, Users may put the "RT" symbol at the beginning of tweet to rewrite other user tweets, then mention the user, followed by a colon, and tweet content. Twitter lets users retweet their tweets as well as those of other users [41]. The retweet network is a directed weighted network; nodes describe users, and edges reflect the retweet relationship. The direction of an edge refers to the direction of information spreading, while the edge weight is how many times a user retweets another user's tweet [42].

A mention is referenced username of another individual anywhere in the body of the tweet preceded by the "@" symbol (i.e., Hi "@John"). Mentions are frequently used to ask users questions, express respect for someone. The profile of a mentioned user is updated. A mention network is a directed weighted network; nodes describe users, and the edges reflect the mentioned relationship. The direction of an edge refers to the direction of user interaction, while the edge weight is how many times a user mentions another user's name in the tweet body.

A reply is a user's ability to respond to a tweet of another user. It's one of the simplest ways to join in on a Twitter conversation as it occurs. The majority of tweets received at least one reply. The reply network is a network the nodes define users, and the edges reflect reply occurrences at a specific period. The individual can respond to a tweet by clicking the prompt above the tweet. Usernames will not be appended at the beginning of the tweet, allowing the user to use all of the permitted characters in your response.

Twitter has many user network attributes used to create different kinds of networks that help enhance social network analysis. Twitter is very noisy content and messy that require a preprocessing phase at the beginning to extract relevant words when analyzing. Tweets have several mistakes, vague terms, numbers, emoticons, unnecessary characters, isolated nodes (vertices), and edges. These difficulties have made using traditional social network analysis methods to find communities on Twitter more challenging. The researchers analyze Twitter social networks in three approaches based on what kind of information is used to study. Twitter can analyze tweet content, the network structure, or their combination. As a result, considering these three types of analysis: content analysis, structural analysis, and hybrid analysis are described below [43][44].

2.3 Content Analysis of Social Network

Social Networks content is easily understandable by users in the natural world, but it is challenging to automatically interpret text, images, and video by machine to extract these data. Users' text contains ambiguity of

words, unstructured text, slang words [45]. This type of content analysis requires the use of machine learning algorithms to convert unstructured text into structured. Natural language processing is one of the most prosperous research areas in this context. Content analysis algorithms were split into three ways: supervised, semi-supervised, and unsupervised learning.

Supervised algorithms need training before they are employed. First, the training data are annotated by humans. Then, supervised algorithms are trained with this labeled dataset. The labeling datasets is often manual and may need professional people to do it. The social network has become an essential source of extracting people's sentiment. Supervised sentiment analysis is a quick and straightforward technique to learn about people's feelings, opinions, and actions to any product, incident, or event. Users' can be classified text into subjective and objective approaches. The subjective process refers to any opinion, review, or discussion, while the objective process includes any neutral content based on facts. The goal of sentiment analysis is to categorize subjectivity-related text into several domain groups [46]. People can discover their interest using sentiment analysis on comments and posts collected from social networks sites. A sentiment analysis method is prepared with the help of labeled data to predict the negative, positive, neutral sentiment. The supervised sentiment methods are usually used to process large datasets. However, manual labeling text for sentiment classifier is extremely difficult and time-consuming. Although collecting a large dataset from social networks has become more available for sentiment analysis, labeling these records remains challenging, limiting the actual size of sentiment analysis.

Supervised classification methods are usually used on Twitter, such as Naïve Bayes, Logistic Regression, and SVM [47][48].

In other situations, labeling the dataset is even impossible. Unsupervised algorithms, unlike supervised algorithms, do not require labeled training data. It learns from the dataset to discover the most common rules that apply to all data. The size of Twitter content is vast due to each tweet is a separate document. Therefore, reading separately all tweets is not possible. These tweets are none labeled, which need an unsupervised approach topic modelling to find a topic [49]. The tweets usually contain a mix of topics that each subject consists of a set of words, and each tweet consists of a group of topics. Topic model is a statistical procedure for identifying latent topics in a set of documents [50]. It uses a text-mining mechanism for finding latent semantic topics in a text [51]. Topic models usually work by grouping documents with similar words in a matching set of documents [52]. The group documents by their dominant topics, with each topic containing a list of keywords. The topic model should produce separate topics that humans can understand. Topic words should all relate to the same cohesive subject with minimal overlap with other topics' concepts. Topic modelling is used to determine which topics are present in the dataset. It creates a probabilistic mixture of words representing word co-occurrence patterns. Various research has studied topic methods to extract topics. For example, Latent Semantic Analysis (LSA) is one of the oldest algorithms that exploit the relationship between documents and terms using a term-document matrix. LSA uses the singular value decomposition (SVD) [53]. Tweets on Twitter are frequently short and contain a lot of unnecessary characters and

misspelled words. These tweets can result in a small number of overlapping words in the set of tweets. As a result, it becomes challenging to determine text-based semantic relations for topic detection.

Semi-supervised learning has much interest in content analysis studies because it can minimize the demand for costly labeled data. Many unlabeled data and a few numbers of labeled data were available online. [54]. A semi-supervised strategy is beneficial when labeled data is limited and unlabeled information is readily available. Semi-supervised algorithms learn from both label and unlabeled datasets to build rules and classifiers [55]. The purpose of semi-supervised learning is to find out how combining labeled and unlabeled data impacts behavior and to develop algorithms that take advantage of it. Semi-supervised learning can also be used as a quantitative technique to study human category learning when most of the input is unlabeled [56]. Content analysis methods are discussed in turn in the following subsections.

2.3.1 Preprocessing

Frequently, a social network text contains some meaningless words. It is often assumed that these words do not affect the meaning of the text. For example, Twitter will have irrelevant words, emaciation, symbol, numbers, emails, and websites links. They exist in part to meet linguistic requirements and user requirements. Before any content analysis can be done, it is challenging to discover topics or sentiments from a collection of posts on social networks using topic modelling and sentiment algorithms because of unstructured text. Before content analysis, these terms should be eliminated from tweets to avoid unexpected errors and

save computational resources. Therefore, posts are cleaned by converting the unstructured text into a more understandable structured format for analysis. The preprocessing procedure is cleaning stopwords, stemming, lemmatization, and tokenizing.

All hyperlinks and non-alphabetic characters like punctuation and digits were eliminated from each text throughout the cleaning process. The text was then be separated by space to produce the list of words. The text was changed into a bag-of-words (BOW) corpus, in which each text is divided into a list of the words. The sequence of words in a BOW corpus is unimportant [49]. Further, tweets contain many stop words and famous English words that have no purpose or meaning, such as articles, prepositions, and punctuations. The preprocessing ensures that the topics and classification produced by topic modelling and sentiment analysis are meaningful and not dominated by the exact top words.

The stemming process is the way of grouping a word's many forms into a single representation, called the stem. In other words, stemming is responsible for obtaining root words by eliminating affixes from the text's words or converting the verb into a noun. For example, "running," "runs," and "run " could all be simplified to a single word, "run". After eliminating the prefix, suffix, insertions, and combinations of prefixes and suffixes, the stem or root of the term remains [57]. In contrast to the steaming process, lemmatization examines the entire vocabulary when applying morphological analysis to words. In other words, The process of collecting together a word's inflected forms. For example, "running" could

all be simplified to a single word, "good". Lemmatization is often considered more informative than stemming.

Tokenizing is the process of separating a sentence into single words (tokens), which can be words, symbols, phrases. The sentence in the analysis is divided into several parts, each of which will contain the words that make it up. The purpose of tokenization is to analyze the words in a sentence. The token list is utilized for other processing as text mining or parsing [57]. Tokenization is helpful for lexical analysis. At first, text data is just a string of characters. The words in the dataset are required for all content analysis operations. As a result, a parser is required for document tokenization. It rapidly turns an unstructured text into structured data suitable for content analysis. Example of sentence " Life consists of decisions and each one you make creates who you are" the word tokenization are " Life ", "consists", "of ", "decisions", " and ", "each ", "one ", "you ", "make ", "creates", "who", "you", "are".

Part-of-Speech (POS) is the process of classifying words into their lexical units and identifying them correctly. Lexical classification or word classes are terms used to describe this procedure. These tags assign each word a lexical group depending on its syntactical discourse and purpose. The English tag signifies whether the word is a noun, adjective, verb, pronouns, conjunctions, interjections, adverbs, and prepositions. It's more challenging to detect part of speech tags than to assign words to their part of speech labels. As a result, POS labeling is not easy work. Sentences contain a variety of tags. As a result, having a consistent mapping for POS tagging is unacceptable [58]. For example, in the sentence" Baghdad is a beautiful city," the tag will be (Baghdad, NNP), (a, DT), (is, VBZ),

(beautiful, JJ), (city, NN). The purpose of the preprocessing stage is to increase the analysis accuracy.

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

TF-IDF has been popularly used for word or phrase extraction. It is a numerical technique used to assess the importance of a term to a document in a collection of documents by multiplying two measured: term frequency and inverse document frequency. As a result, terms like (if, who, and that) often appear in all documents, have fewer scores because they don't mean anything in the document. Nevertheless, if the term (happy) frequently occurs in one document but not in others, it can be very important. The term (happy) will undoubtedly be related to the subject's dependability.

A TF-IDF is a mathematical calculation that the term frequency counts how many times a term occurs in a document. The term frequency can be modified based on the size of the document or the raw probability of the most commonly utilized term in the document. The inverse document frequency of the term over a set of documents indicates how frequent or rare a term is among whole document collection. The value is near zero when the word is regular in the corpus. The logarithm is divided by the total number of documents by the number of documents having a word. The TF-IDF term score is calculated by multiplying term frequency and inverse document frequency values [59]. The higher value of TF-IDF refers to the critical term is in the document. On Twitter, the TF-IDF score for the term i in tweet j from tweets set D was calculated as in mathematical equations below:

$$TF = \frac{No.of\ reptitions\ of\ word\ in\ a\ tweet}{No.of\ word\ in\ a\ tweet} \quad (2.1)$$

$$IDF = \log \frac{\text{Number of tweets}}{\text{Number of tweets containing the word}} \quad (2.2)$$

$$TF - IDF_{i,j} = TF_{i,j} \times IDF_i \quad (2.3)$$

Where $TF_{i,j}$ refers to the number of occurrences that term i occurs in tweet j, IDF_i is the logarithm of dividing the number of tweet N by the number of tweets containing term i [60]. TF-IDF has been used on different applications like text analysis for data mining [61].

2.3.3 Latent Semantic Analysis (LSA)

LSA is an unsupervised analysis approach to extract the latent semantic structure from a set of documents. Latent means the hidden concept in the data. Deerwster introduced LSA in [10] to improve information retrieval by considering conceptual content rather than matching similar words and reducing the dimensionality of a matrix using Singular Value Decomposition (SVD). LSA is a topic model algorithm for automatically determining hidden topics in a corpus. It describes the meaning of a word as the average meaning of the documents [57]. The meaning of a document is defined by the average meaning of all words in the documents. LSA shows word-word relationships that are closer to human recognition [62]. LSA measures word occurrence count information and inferred essential relations and meanings between words in the document. According to scientific evidence, when people write text, the vocabulary they use reflects the meaning of the text. However, words may have

different meanings in different documents. LSA can extract latent meanings from texts. Traditional clustering algorithms cannot deal with synonymy and polysemy in documents. Synonymy has referred to many terms to guide to a similar topic. Polysemy refers to the concept that a term can carry multiple meanings or relate to various objects [51]. For example, if a user uses the word "automatic" and "electrical", clustering algorithms group only documents that contain words related to "electrical". However, if the document contains the word " automatic", the user may not get the desired outcome. The LSA method deals nicely with synonymy, described as different words or phrases with the same meaning.

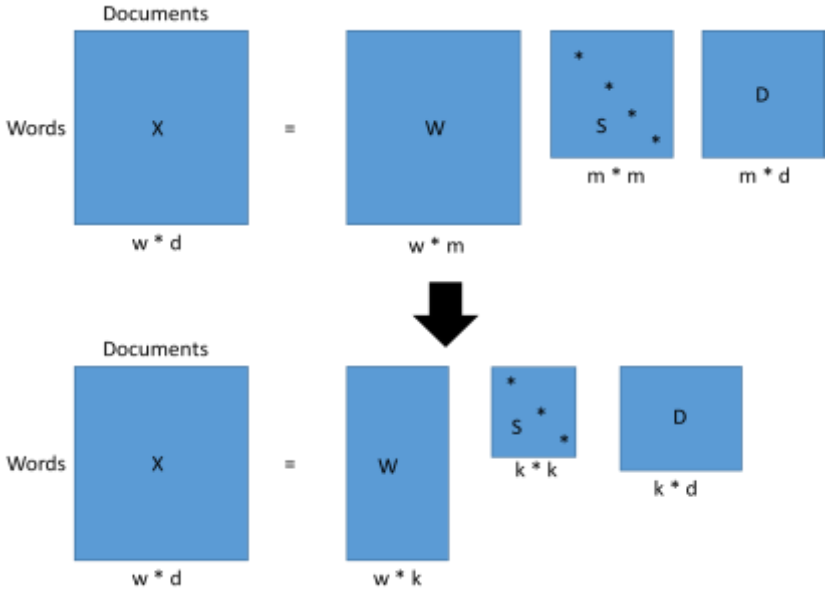


Figure 2. 2 Decomposition Word-Document Matrix [10]

LSA processes the words in the corpus to build a word-document matrix to indicate the link between term occurrences in documents. The rows of a word-document matrix represent words (dictionary), while the columns represent various documents. Dictionary is a set of all words that appear in at least one document in the dataset. The element of the word-document matrix represents the number of times the term occurs in each document [63]. TF-IDF value can replace the word frequency of the word-document matrix to show important documents. This matrix is often sparse, with rows representing words and columns representing documents. The LSA applies the SVD to the word-document matrix that decomposes the original word-document matrix into three matrixes as in Figure (2.2). The orthogonal W, a diagonal S, and a transpose matrix from the orthogonal D are three matrixes. This decomposition only kept the essential dimensions associated with the highest singular values of the co-occurrence matrix that provided a low-rank approximation of the word-document matrix. The rank approximation is to reduce unnecessary data from the original dataset. SVD decomposes this matrix to derive k number of topics. Figure (2.3) shown the LSA steps. The k topics can be obtained by training the topic model algorithm with LSA. a different number of k to identify the best topics. The number of a topic must less than the number of words. Therefore, learning the concrete structure of documents might allow to assess a document at the topic level rather than the word level. SVD is used to reduce matrix dimensionality, representing a semantic matrix to capture the relationships between words and documents. The SVD can be expressed mathematically as an Equation (2.4) [10].

$$X = W_{w*m} \times S_{m*m} \times D^{T}_{d*m} \quad (2.4)$$

It can show the new word-documents matrix of rank k closest to X as an Equation (2.5).

$$\hat{\mathrm{x}} = W_k \times S_k \times D^{T}_{K} \quad (2.5)$$

Where X is the word documents matrix with size w× d (w is the number of words and d number of a document). W and D^{T} represent the rank k reduce matrix with a size equal to t × k and k × d. S is the diagonal matrix of singular values. k is the mount of dimension reduction of matrix.

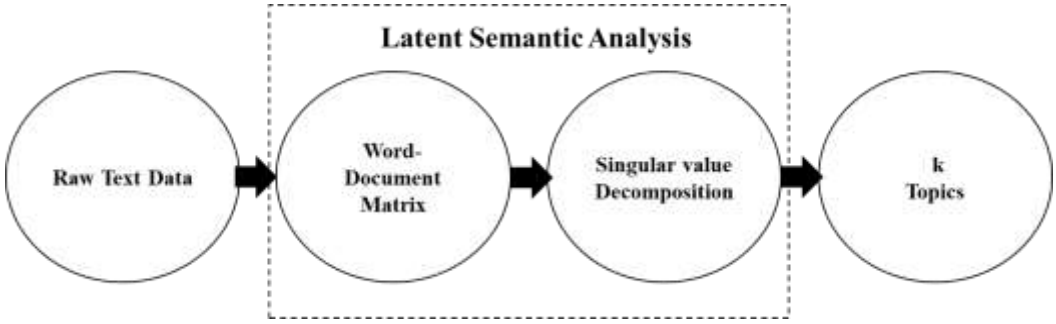


Figure 2. 3 LSA Steps [10]

2.3.4 Support Vector Machine (SVM)

SVM is a supervised classification model that tries to find a proper hyperplane to split the data samples [64]. The principle of classification is to maximize the margin among classes. In other words, it attempts to maximize the partition borders among data points depending on the

classes. The minimum distance between data points and hyperplane are called support vectors. Linear support vector machine is the most fundamental concept to find an appropriate separation hyperplane in the training sample dataset. Since the maximum margin is desired, it is crucial to compute the margin in the sample area. The following equation illustrates the division of the hyperplane [65]:

$$W^T \mathrm{x} + \mathrm{b} = 0 \tag{2.6}$$

Where W represents a standard vector that specifies the orientation of the hyperplane. Displacement represents by b defines the distance between the origin and the hyperplane. Assume the hyperplane can correctly classify the training data using the following formula:

$$\begin{aligned} W^T \mathrm{x} + \mathrm{b} &\geq 1, \qquad y = 1 \\ W^T \mathrm{x} + \mathrm{b} &\leq -1, \qquad y = -1 \end{aligned} \tag{2.7}$$

Where y = 1 for positive classification, and y = -1 for negative classification. SVM used training data $\{t_1, \ldots, t_n\}$ which represent vectors in space and labels $\{L_1, \ldots, L_n\}$ where $L_i \in \{1, 0, -1\}$ to train classifier. SVM hyperplanes split the training data into a maximal margin among classes. The training samples nearest to the hyperplane are named support vectors. SVM classifier trained with the labeled dataset can give labels to the unlabeled data by discovering the hyperplane, which maximizes the margin among data (see Figure 2.4).

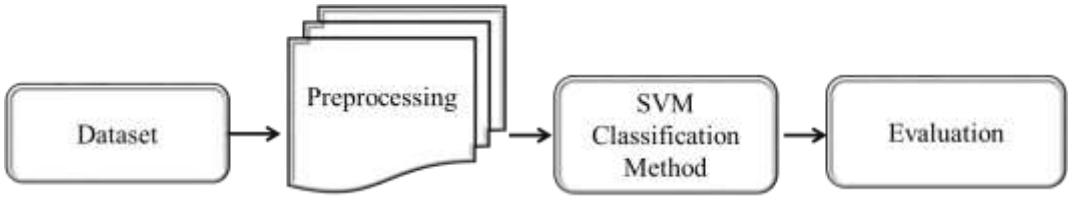


Figure 2. 4 SVM Method [64]

2.4 Structural Analysis of Social Network

In recent years, structural social networks have been emerged for visualizing and analyzing networks relationships. Researchers have found the importance of social network analysis in many areas [36] such as biology, network studies, business, knowledge science, public health, computer science. Structural networks used graph theory to analyze relations in networks. Their main methods and techniques have been applied in large social network problems (i.e., Twitter, Facebook, Instagram, LinkedIn, Snapchat, YouTube). Social networks are considered complex networks because of the sparse and noisy network topology of social networks. In other words, node-to-node connections will not be random or completely regular [40].

Structural social network analysis refers to a method for extracting meaningful information from networks regarding their structures by measuring the relationship among network participants, including users, companies, URLs, or any other type of connected information processing structure [66]. Structural network analysis aims to detect similar patterns in the network, where the structure network plays as important as the data

contents. It has been utilized to extract meaningful information from complex social networks. For example, the Twitter network has been used to find the influencer property, which is the property of finding out the unique users' match network in terms of the number of connectivity, degree centrality, and degree of betweenness concerning the nodes in a network. The structural network involves identifying the most influential, famous, or central individuals using link analysis methods, communities are discovered using community detection methods; and information propagates through the network using diffusion algorithms. These analyses are particularly useful in extracting information from networks and, as a result, in the problem-solving process.

The structure of interactions between social entities is investigated in social network analysis. Relationships established can be personal or professional and can range from casual acquaintance to close familiar links. Links can reflect the flow of information, interactions, and similarities. Graphs are commonly used to illustrate the structure of such networks. A graph is comprised of two essential elements: vertices and edges. Graph theory describes social networks as vertices connected by edges, with vertices representing persons and edges representing their connections [66]. Depending on the application need, vertices can represent a wide range of individual objects such as (humans, governments, publications, companies, businesses, plants, and animals). On the other hand, an edge is a line that links two vertices and can represent a variety of interactions between distinct objects (e.g., friendship, communication, cooperation, acquaintances, and trade). Edges

might be directed or undirected depending on whether the relationship is asymmetric or symmetric. A graph (network) is represented a non-empty collection *V* of vertices and edges, where *E* is the number of edges defined as $G = (V, E)$.

The network can be a static structure when has links between users are stable, such as the relationship of a friend or follower links. These relationships have continued for an extended period. Nevertheless, people do not depend on their friends' connections but interact with them. According to Twitter data, most interact with users with and who they do not share followership connections. A small percentage of users actively use followership links to communicate with one another. The network structure can be dynamic depending on the interaction between users because the links only exist if the users have connected in a short time [67]. The interactions relationship provides more significant information and reflects the power of connections between users. These may typically be defined by assigning edge weights to the interactions they measure, which better integrate and improve them. Community detection focusing on interaction structure rather than static structure will increase the accuracy of the measure of cohesion and discover more relevant groups.

Most community detection literature presents graphs as input and focuses on finding a cohesive structure (communities) dependent on user interactions. Community detection is a critical problem on social networks. It likes a graph theory that partition a graph [20]. Several researchers have tried to identify optimal clusters of users. Community detection is the process of detecting groups connected closely within a

network's structure. It may be a helpful tool to find community because the communities can be a compressed version of the large network to get a whole picture of the network. Modularity maximization [68] is a common type of community detection method. It aims to maximize the modularity of the cluster allocation, represented as the quality of partitions defined by the variation number of edges inside a community. The Louvain method [6] greedy update adjusts node by node and uses the best neighboring community to optimize the modularity function gain. The algorithm then aggregates the result partition and repeats the process until no new communities are developed. The Louvain technique is fast and efficient, but it still gets stuck at local optima and might cause communities to become separated. The Leiden method [17] resolves the Louvain problem by adding a refinement step, but it still depends on greedy local updates and is vulnerable to local optima.

Community detection is a process of partitioning a network into clusters of closely connected nodes. In other words, nodes in the same community should have a high connection to each other while having a low connection to nodes in other communities. Various communities detections methods are used, such as the Louvain algorithm and Leiden algorithm [69]. These algorithms are most appropriate for identifying communities in large networks, which are common in social network data. Discovering communities in a social network can help many target businesses, marketing, information propagation, and online purchases recommendations.

However, the complexity and diversity of networks are often hard to distinguish in real datasets networks. As a result, algorithm design has become a critical issue for many networks. Community detection methods could have an immediate influence on network organization and function. The novel cluster agglomerate algorithm (CAA) similarity index proposed incorporates local information, and the criterion is a global index that conforms to the community's ideal state. The divide and agglomerate (DA) algorithm is accomplished by findings a two-step method: splitting a network into small clusters based on the similarity of node connections and combining a cluster with the one that has the most interest for it until the community requirement is stable [70]. A novel multi-objective community detection method established multi-objective particle swarm optimization (MOPSO) has been proposed [71]. The Pareto dominance strategy was used to solve the graph clustering problem by minimizing two goals, Ratio Cut and Kernel K-Means. The moving process of particles changed by taking the crossover operation to improve its local and global best. The following subsections introduce the most popular ways of finding communities in social networks.

2.4.1 Louvain Method

Blondel et al. [6] created the Louvain algorithm, a frequently used community discovery method for large-scale networks, a popular greedy approach for community discovery. It is based on modularity maximization. Louvain method has been widely used in multiple application areas [72][73] because of its fast convergence features, hierarchical partitioning, and high modularity. The method maximizes

each community's modularity score between −1 and 1, which measures the distribution of nodes in groups by comparing the density of connection among nodes to how they can link in a random network.

Louvain algorithm was divided into two stages: nodes moving locally and network aggregation. Those are repeated iteratively to increase the modularity value. Individually node of the network is assigned into a different community which each node is its community. The difference in modularity is determined in each community by moving nodes between communities (i.e., node i removing from its cluster and putting it into the adjacent cluster j). However, this modularity must be positive. If no positive result can be achieved, i will return to its original community. The method performs two computations for each node i. The modularity gain ΔQ was computed when placing node i in the cluster of any neighbor j. Then, choose the community that offers the biggest modularity gain and join the appropriate community as Equation (2.8). This process is repeated until no gain results change. The modularity ΔQ can be computed by the equation below. This process is continued and successively applied to all nodes until no more improvement can be made, at which point the first phase ends [6][74].

$$\Delta Q = \left[\frac{\Sigma_{in} + \Sigma_{i,in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \quad (2.8)$$

Where Σ_{in} the aggregate of the weights of the connections in cluster C (weight of internal edges), Σ_{tot} is the aggregate of the weights of the connections occurrence to nodes in cluster C (weights of all edges

connection in community), $k_i$is the aggregate of the weights of the connections occurrence to node i, $k_{i,in}$ is the aggregate of the weights of the connections from i to n nodes in cluster C, and m is the aggregate of the weights of all the connections in the network.

After local moving in the previous stage, the aggregation stage of the method presented nodes of the community by one node and edges by a self-loop. The weight of the self-loop is the number of edges. Connections among communities are compressed by one connection, with weight representing the number of links between its nodes and the nodes of each community. After the second step, the method loops back to the first step and uses the two-stage procedure on the graph generated in the last iteration (communities exchange the nodes). This procedure is repeated until the modularity value for each node in the graph gets the maximum.

As a result, the node would be assigned to a single community (see Figure 2.5). In other words, this algorithm seeks for small communities by maximizing modularity locally. Then continues the process until the maximum modularity is obtained. This strategy does not specify the number of communities to be discovered but instead builds hierarchical communities from the bottom up (see Algorithm 2.1) [17]. The modularity optimization-based community detection technique has a significant limitation in that it cannot find communities less than a specific size. Therefore, defining the resolution in the method is critical, as it determines the size of the smallest community to be discovered [75]. Lower the resolution value can get more communities (smaller communities), and a higher resolution value can get fewer communities (bigger communities).

Algorithm 2. 1 Louvain Algorithm [17]

```
1.  function Louvain(Graph G, Partition P)
2.  do
3.      P ← MoveNodes(G, P)          //Move nodes between communities
4.      done ← | P| = |V(G)|         //Terminate when each community consists
                                       of only one node
5.      if not done then
6.        G ← AggregateGraph(G, P)   //Create aggregate graph based on partition
7.        P ← SingletonPartition(G)  //Assign each node in aggregate graph to
                                       its own community
8.      end if
9.  while not done
10. return flat*(P)
11. end function

12. function MoveNodes(Graph G, Partition P)
13.    do
14.      H_old = H(P)
15.      for v ∈V (G) do              //Visit nodes (in random order)
16.          Ć ← arg max_{c ∈p ∪∅} Δ H_p(v → C)  //Determine best community for
                                                    node v
17.          if Δ H_p(v → Ć)> 0 then   //Perform only strictly positive node
                                         movements
18.              v → Ć                 // Move node v to community Ć
19.          end if
20.      end for
21.    while H(P) > H_old             // Continue until no more nodes can be moved
22.    return P
23. end function

24. function AggregateGraph(Graph G, Partition P)
25.     v → P                         // Communities become nodes in aggregate graph
26.     E ← {(C, D)|(u, v) ∈ E(G), u ∈ C ∈ P, v ∈ D ∈ P)  // E is multiset
27.     return GRAPH (V,E)
28. end function

29. function SingletonPartition(Graph G)
30.     return {{v}|v ∈V(G)}          //Assign each node to its own community
31. end function
```

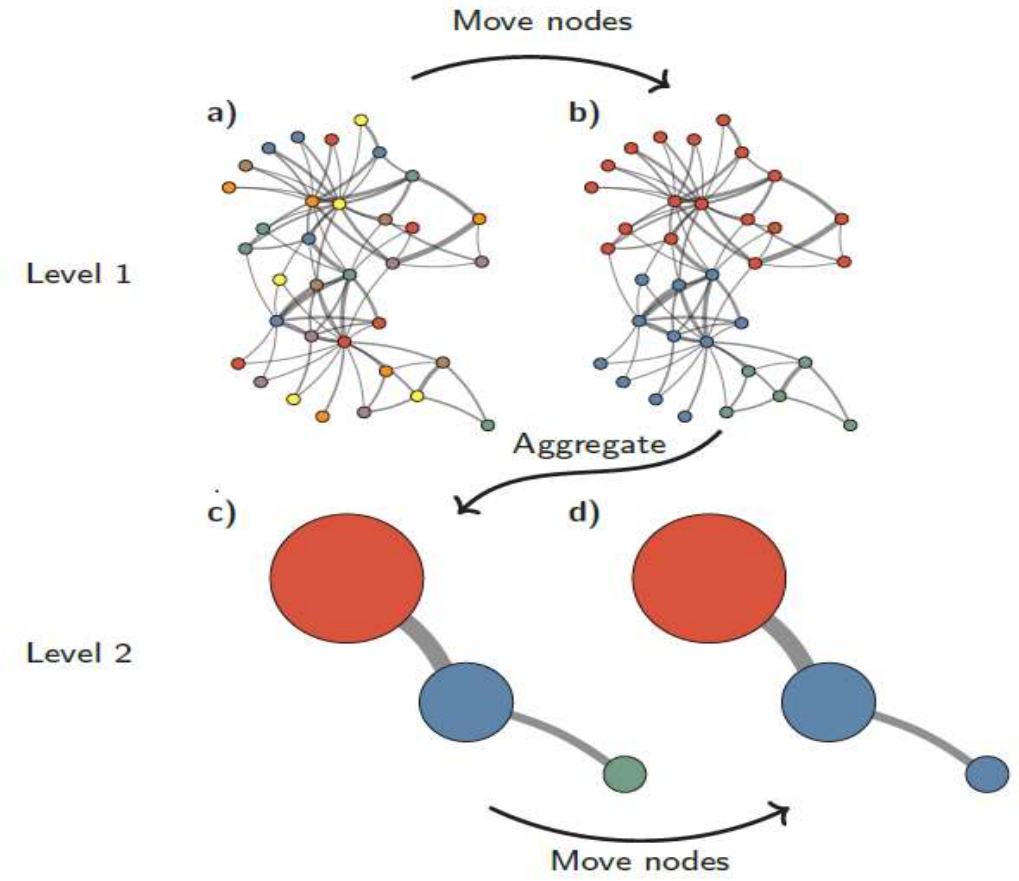


Figure 2. 5 Louvain Method [17]

2.4.2 Leiden Method

Detecting the community is significant work when investigating complex social networks. Leiden was built based on the Louvain algorithm that overcomes the problem of community connection badly. The Louvain method may cause to discover communities that are arbitrarily badly connected. It can detect internally disconnected communities where one part of the community only connects with another piece of the same community by the external edge. In Figure (2.6), figure

(a) shows that the red community becomes internally disjointed when node 0 moves to a different neighborhood. At the same time, figure (b) demonstrated that Louvain keeps nodes 1–6 locally optimally allocated in the same community even though node 0 moved into another community.

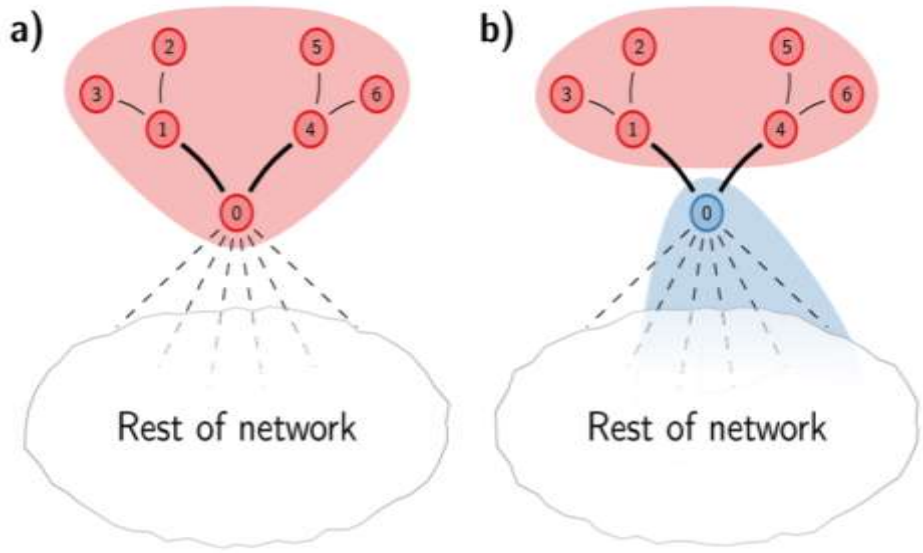


Figure 2. 6 Louvain Arbitrarily Poorly Connected [17]

Leiden is an iterative method that guarantees nodes inside communities are strongly connected, and all subsets communities are assigned optimally [17]. The Leiden method is established partly on the smart local move method, which is a modification of the Louvain method. It also uses ideas of speeding up local node movement and moving nodes to random neighbors. These are the most helpful ways to enhance the Louvain algorithm. The Leiden algorithm is simple to understand and easy to implement. It has less time complexity than the Louvain algorithm. It consists of three stages: nodes moving locally, refinement of the separation, and network aggregation based on refined split.

The Leiden algorithm's initial stage starts with a community of singleton nodes. The nodes then move across communities in seek of the optimal partition. The algorithm then performs a refined stage that aggregates the network. The Leiden algorithm utilized a fast local movement algorithm that only visited nodes whose neighborhood had changed. The Louvain approach, on the other hand, repeatedly examines each node in a network. The refinement stage might split a community into multiple communities when they have poorly become connected after a node move, increasing the connectedness of the remaining communities. Nodes are not always greedily combined with the cluster that provides the most significant gain. A random cluster is selected to merge nodes with it, and the quality function is calculated. The cluster with higher quality function value is chosen. The Leiden stages showed in Algorithm (2.2).

Figure (2.7) shows all steps on the Leiden method begin from (a) a singleton community. Then step (b) moves individual nodes between communities to find the best partition, and step (c) refines communities. Then (d) aggregate network based on refined step. For example in (b) the red community is refined into two sub groups in (c), which, after aggregation, convert into two isolated nodes in (d), both belonging to the same community. The method then moves separate nodes in the aggregate network (e). In this network, refinement does not change the partition (f). Recently, the Leiden algorithm has been used widely in many application biological areas [76][77] because of its guaranteed connection, hierarchical partitioning, and high modularity.

Algorithm 2. 2 Leiden Algorithm[17]

```
1.  function LEIDEN(Graph G, Partition P)
2.    do
3.      P ← MOVENODESFAST(G, P)        ///Move nodes between communities
4.      done ← | P |= V (G)            //Terminate when each community consists
                                          of only one node
5.      if not done then
6.        P_refined ← REFINEPARTITION(G, P)        //Refine partition P
7.        G ← AGGREGATEGRAPH(G, P_refined)  //Create aggregate graph based
                                          on refined partition P_refined
8.        P ← {{v | v ⊆ C, v ∈ V (G)} | C ∈ P}     // But maintain partition P
9.      end if
10.   while not done
11.  return flat∗( P )
12. end function

13.  function MOVENODESFAST(Graph G, Partition P)
14.    Q ← QUEUE(V (G))  //Make sure that all nodes will be visited (in random
                                          order)
15.    do
16.      v ← Q.remove()                  // Determine next node to visit
17.      Ć ← arg max_{c ∈p ∪∅} Δ H_p(v → C)   //Determine best community for
                                          node v
18.      if Δ H_p(v → Ć)> 0 then        //Perform only strictly positive node
                                          movements
19.        v → Ć                         // Move node v to community Ć
20.        N ← {u |(u, v)∈ E(G), u ∉ Ć    //Identify neighbours of node v that are
                                          not in community Ć
21.        Q.add (N − Q)          //Make sure that these neighbours will be visited
22.      end if
23.    while Q ≠ ∅                  // Continue until there are no more nodes to visit
24.  return P
25.  end function

26.  function REFINEPARTITION(Graph G, Partition P)
27.    P_refined ← SINGLETONPARTITION(G)      //Assign each node to its own
                                          community
28.    for C ∈ P do                          // Visit communities
29.      P_refined ← MERGENODESSUBSET(G, P_refined, C)//Refine community C
30.    end for
```

31. **return** $P_{refined}$
32. **end function**

33. **function** MERGENODESSUBSET(Graph G, Partition P, Subset S)
34. R = {v | v ∈S, E(v, S − v) ≥ γ ‖ v ‖.(‖ S ‖−‖ v ‖) //Consider only nodes that are well connected within subset S

35. **for** v ∈ R do // Visit nodes (in random order)
36. **if** v in singleton community **then** //Consider only nodes that have not yet been merged
37. $T \longleftarrow \{C \mid C \in P, C \subsetneq S, E(C, S - C) \geq \gamma \parallel C \parallel \cdot (\parallel S \parallel - \parallel C \parallel)$ //Consider only well-connected communities
38. Pr(Ć = C) ~ $\begin{cases} \exp(\frac{1}{v}\, \Delta H_p(v \rightarrow C)) & if\ \Delta H_p(v \rightarrow C) \geq 0 \quad \text{for } C \in T \\ 0 & \text{otherwise} \end{cases}$ // choose random community Ć
39. v → Ć // Move node v to community Ć
40. **end if**
41. **end for**
42. **return** P
43. **end function**

44. **function** AGGREGATEGRAPH(Graph G, Partition P)
45. V← P // Communities become nodes in aggregate graph
46. E← {(C, D) | (u, v)∈ E(G), u ∈ C∈ P, v ∈ D ∈ P} //E is a multiset
47. **return** GRAPH(V, E)
48. **end function**

49: **function** SINGLETONPARTITION(Graph G)
50: **return** {{v}| v ∈ V (G)} // Assign each node to its own community
51: **end function**

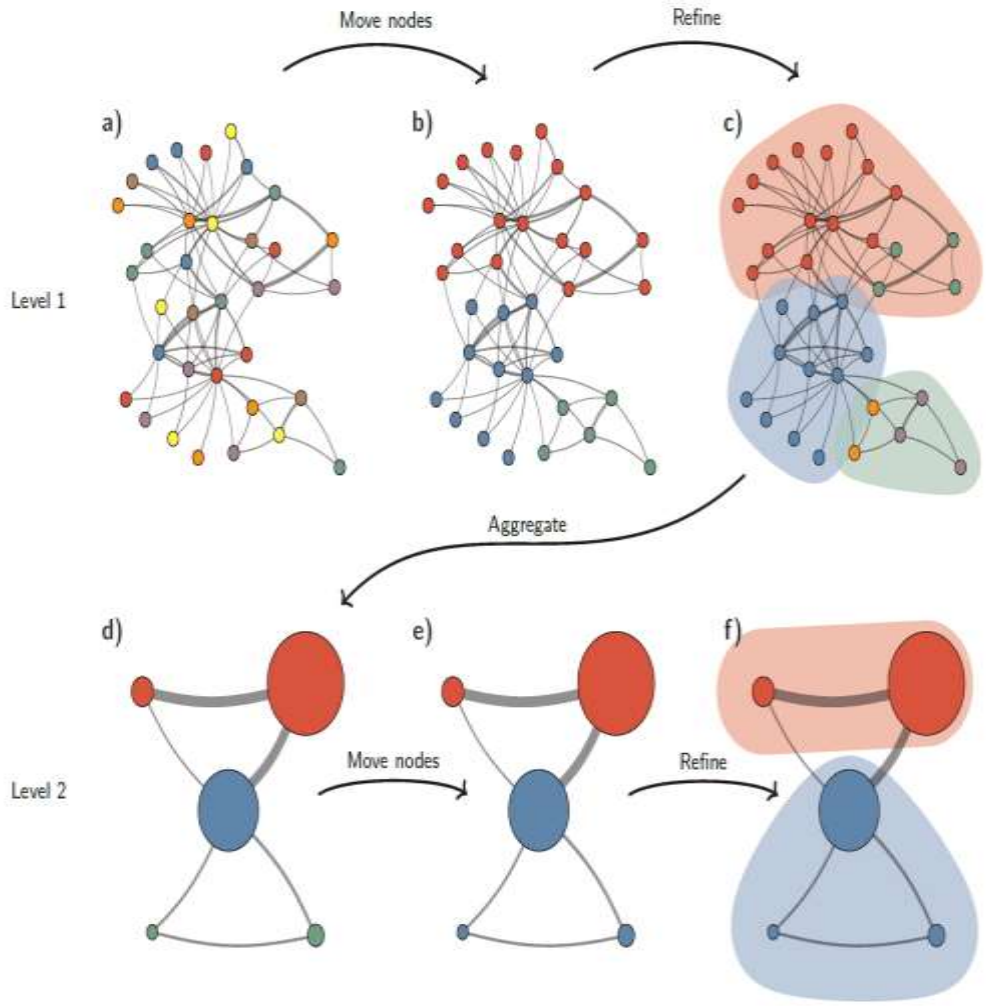


Figure 2. 7 Leiden Method [17]

2.5 Hybrid Analysis of Social Network

A variety of approaches might be applied to analyze social networks. Specific methodologies are more suited depending on the research emphasis (e.g., text, interactions, topics, opinion). Several researches have focused on discovering sentiment analysis and topic extraction using standard methods for analyzing content social network datasets [78].

Various current topic models have been applied for document analysis to find the latent topics from text such as LDA, LSA, and PLSI. Topic model approaches have successfully derived topics for the lengthy document in the traditional technique. For example, a study [79] introduces latent Dirichlet allocation is based on probabilistic topic model. They have presented two experiments: topic modelling of Wikipedia pages. The document topic method created to provide a topic-based explanation for finding, browsing and suggesting papers. The authors in a study [80] proposed six different preprocessing methods that affect sentiment polarity classification in Twitter. However, topic methods suffer when used to identify latent topics in a social network such as Twitter.

On the other hand, social networks are crucial to understanding how people and groups interact. Social network analysis is helped study these relationships. It analyzes the structural network by focusing on extracting common network features or user network features. For example, finding communities and influencers in the network can be helpful approaches in social network analysis, such as community detection algorithms. Structure network analysis only focuses on nodes relationships to find communities that neglect the other network features such as content which may cause poor quality results.

Recently, many studies proposed combing community detection algorithms with text mining methods. These studies tried to incorporate users content into the structural network. In other words, combining structural analysis with content analysis techniques provides a new approach to discovering communities. Some research has taken advantage

of social network features to improve topic extraction in short text like Twitter. W. Cui et al. [81] introduced combining social hashtag network feature and short text, which increase each text by associated hashtags to get complete semantic features. Prateek et al. [82] suggested a new pooling method by incorporating a community detection method to update the topic derivation without changing the LDA topic model. This approach was aggregated user text content that shares topics and relations. Other research has tried to enhance the quality of community detection algorithms by incorporating text and other features into the weight of edges. Li et al. [27] developed a hierarchical clustering method on user relationships and interests networks to discover the communities. Semantic analysis and the following/follower relationships are used to calculate edge weight to improve community detection. Chunaev et al. [83] presented an integrated weight model that exploits multiple existing weights on the social network. It investigates several models that combine between structure and attributes of networks.

2.6 Evaluation Metrics of Social Network Analysis

Social networks analysis uses different methods to complete the investigation. Therefore, different evaluation metrics are used with content and structure social network analysis. For content analysis, topic modelling algorithms are automatically determining latent topics in documents corpus. However, evaluating such assumptions is challenging due to its unsupervised techniques. Topic models have been assessed using a coherence score. Topic coherence can be described as the degree of connection among the words inside a topic [84]. The coherence metric

is utilized to evaluate the degree of semantic similarity among words on the topic. The coherence score considers a topic by calculating word co-occurrences and mutual information to indicate in what way individuals understand the issue. It includes the degree of semantic similarity between words on the topic. The coherence score is a value representing the topics model is good or not, which the higher value indicates the good topics. The coherence score is based on a sliding window, normalized pointwise mutual information (NPMI), and the cosine similarity. It evaluates the quality of coherence of words in the topics as the following equation.

$$coherence(V) = \sum_{(vi,vj)\in V} \text{score}\ (\text{vi}, \text{vj}, \in) \qquad (2.9)$$

Where V is a set of words that describe the topic and $\in$ indicates a smoothing factor, the score will always return real numbers. The UCI and UMass measures calculate the coherence of a topic as the aggregate of pairwise distributional similarity scores over the set of topic words. UCI is a word pair's value is the pointwise mutual information (PMI) between two words as Equation (2.10) [85].

$$\text{score}\ (\text{vi}, \text{vj}, \in) = \log \frac{P(v_i, v_j) + \in}{P(v_i)\ p(\ v_j)} \qquad (2.10)$$

UMass is the score to be based on document co-occurrence as Equation (2.11).

$$\text{score}\ (\text{vi}, \text{vj}, \in) = \log \frac{D(v_i, v_j) + \in}{D(\ v_j)} \qquad (2.11)$$

Further for sentiment analysis, the quality of the classification result is measured using precision P, recall R, F-score, and accuracy [86]. Precision (P) is the proportion of expected positive occurrences to the total number of positive occurrences. It's calculated as follows:

$$p = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2.12)$$

A recall (R) is the proportion of perfectly predicted positive occurrences to total occurrences in the actual class. It's calculated as follows:

$$R = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2.13)$$

The F-score is a weighted average of recall and precision. When data has an uneven class distribution, precision becomes less relevant. It's calculated as follows:

$$F - score = 2 * \frac{P * R}{P + R} \quad (2.14)$$

Accuracy calculates the ratio of the total number of correct predictions., as seen in the equation below.

$$Acc = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (2.15)$$

On the other hand, structure analysis has different evaluation metrics. Community detection is a difficult task due to large members in the community that are rarely known. The accuracy of the social network clustering methods relies on the capability to discover the number of

clusters in the networks, where the networks structures are known. Various community metrics help to evaluate the quality of a community. The evaluation is dependent on the community network structure as modular, dense, and balanced. Two popular measures are used to calculate community detection efficiency called modularity and Constant Potts Model (CPM).

Modularity is one of the essential measurements to recognize good partitions for community detection. It is established as a density metric for the proportion of links within the community compared with the other links allocated randomly within the same community. Modularity measures the variation between the current number of edges in a group and the expected number of edges in a random graph. It tries to maximize their difference [34]. The modularity values are calculated using the Equation (2.16).

$$Q = \frac{1}{2m}\sum_c \left[e_c - \gamma\left(\frac{K_c^2}{2m}\right)\right] \qquad (2.16)$$

Where c represents community, e_c denotes real no. of edges in community, m indicates the total number of edges in the dataset. $^{K_c^2}/_{2m}$ indicates the expected no. of edges and K_c denotes the aggregate of degrees of nodes in each community. γ denotes resolution parameter. Modularity suffers from resolution parameters preventing small communities' detection in a large network.

The Constant Potts Model (CPM) is the quality function which overcomes resolution limitations of modularity. It partitions communities

into two communities dependent on value if the link density between them is lower than the constant value. CPM attempts to maximize the number of inner edges while at the exact timekeeping relatively small communities. The parameter γ balances these two constraints. It serves as the outer and inner edges density threshold. Therefore, the parameter performs as a resolution that the higher value will cause fewer communities. The CPM values are calculated using the Equation (2.17) [17].

$$Q = \sum_c [e_c - \gamma(n_c^2)] \qquad (2.17)$$

e_c denotes real no. of edges inside community and n_c represents no. of nodes in communities.

Chapter Three

Design and Develop

Community Detection Model

3.1 Overview

This chapter describes the complete development techniques used to enhance the community detection method in social networks. To achieve this, a combination of the methods mentioned in the previous chapter would be a great way to build the proposed model. The proposed model of enhanced community detection is generally illustrated in section 3.2 to show the architecture of the proposed model. The datasets collection is defined in section 3.3. The preprocessing procedure is described in section 3.4. In section 3.5, the mechanism of finding topics is determined, then sentiment analysis is outlined in section 3.6. Finally, community detection is stated in section 3.7.

3.2 Community Detection Model

The proposed community detection model was designed and developed to enhance detecting communities by adding more helpful information on communities rather than only depending on network relationships. It is primarily characterized by incorporating three fundamental stages (see Figure 3.1). Each stage uses a specific method to enhance the community detection algorithm in a social network. The first method is LSA topic modelling. It is applied to users' content in a Twitter social network. Topic model was used to discover hidden topics in tweets and assign them to specific tweets. The second method is a Support Vector Machine (SVM) applied to get users' opinions on tweets. Then, these methods combined with user attributes to get a new weighted graph to improve the quality function of community detection in social networks. Finally, Leiden community detection is enhanced by exploiting a new weighted graph to

improve the quality of modularity and a Constant Potts Model (CPM). Figure (3.2) shows the general framework of our model through the following steps:

1. The preprocessing has several steps (building dataset via Twitter streaming API, filtering and normalizing tweets that convert the unstructured tweets into a more acceptable readable structure, building a network, data extraction).
2. Exploit LSA topic model with hashtag to detect latent topic in tweets.
3. SVM sentiment analysis algorithm used to classify tweets into positive, natural, and negative.
4. A new weighted network is developed by calculating during the community detection step to enhance modularity quality and a Constant Potts Model (CPM) for Leiden community detection.

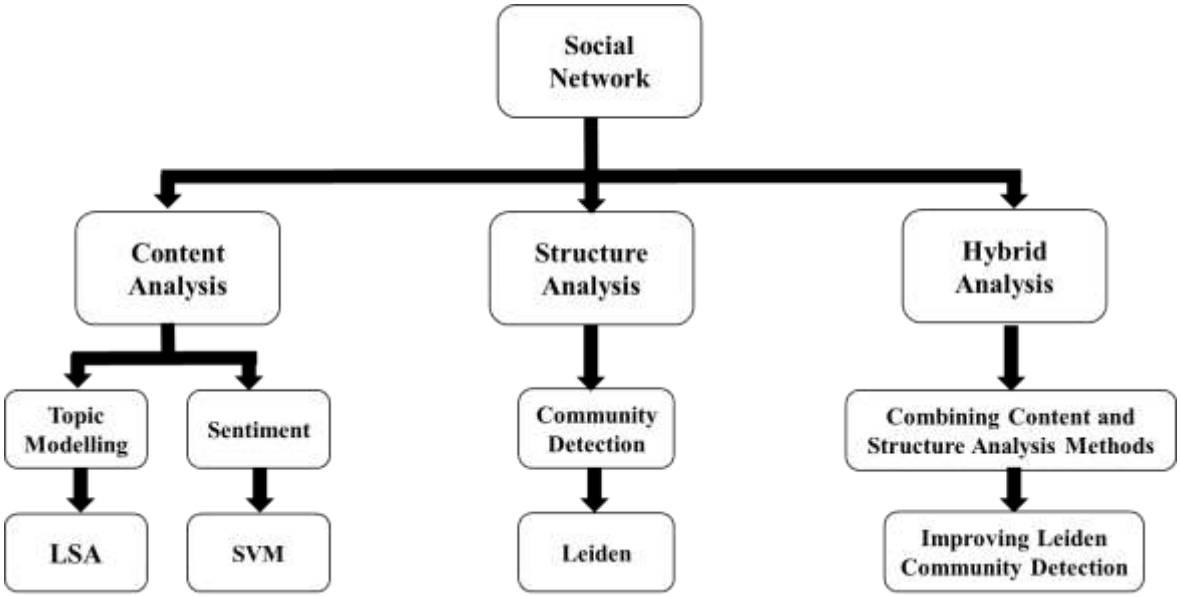


Figure 3.1 Methods for Improve Social Network Analysis

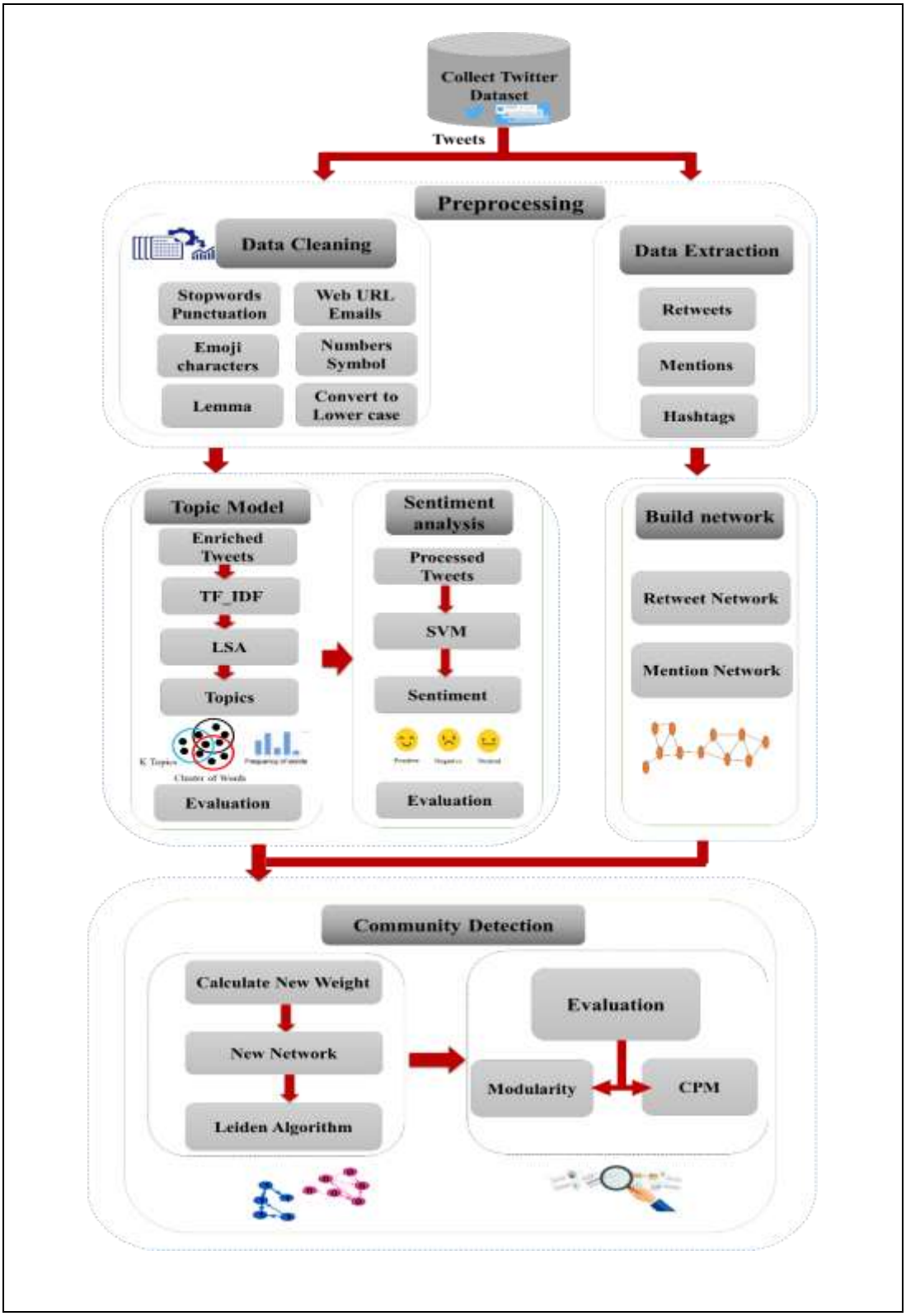


Figure 3. 2 Proposed Model

3.3 Twitter Collection

Twitter provides an API stream platform to accumulate real-time tweets. People share their thoughts about their daily lives with various world topics. Twitter allows people to write 280-character text that is known as tweets. Other people can like, comment, mention other users' names on a tweet or retweet your tweets. People can follow one another on Twitter or become following with other company or news account. Unlike most other social networks, Twitter allows one-way links, meaning that one user can follow another without the other exchanging contact. These interactions used to build a communication network. Several of the studies utilized the API stream as a source of data.

The API stream collects tweets data using a set of predefined keywords and hashtags. Tweets have been collected from Twitter, and they are saved in a JSON file format. JSON data converts key and value pairs to CSV, a basic file format for recording tabular data. The headers for the CSV file will be the keys, and the data collected will be the value. Tweet in Twitter has many attributes like (post time, name, location, hashtag, retweet, mention, reply). Therefore, this model utilized the attributes created_at, id, text, user_name. The two raw data are collected from Twitter APIs using the Algorithm (3.1). It returns tweets information and saves them in CSV files. The first raw data included tweets about education, car, anxiety, entertainment, and other subjects. The second set of tweets was collected about the social media website, phone, and electric devices. These raw data will be formalized by cleaning and organizing them in a dataset format to be utilized in future studies on this topic.

Algorithm 3. 1 Collect Data from Twitter Streaming API

Input: Consumer Secret cs, Consumer Key ck, Access Token at, and Access Secret as // Twitter API access keys

Output: F // CSV Tweets File

Begin

1. Call Twitter API (ck, cs, at, as) //user access account and Twitter access key
2. Languages = ['en'] // select English language
3. Filterss = [keywords, hashtags, Languages] // Query
4. Collect Tweets (Filterss) // From Twitter API stream
5. Save (JS) // JSON file
6. JF = Read (JS) // read JSON file
7. For line in JF do
8. F= Select [Created_at, Id, Text, , user_name, retweeted]
9. F= Convert (F) //Convert select JSON attributes into CSV file
10. End for
11. Return (F) // CSV file

End

3.4 Twitter Data Preprocessing

Preprocessing is an essential part of any model since the words, sentences, and networks identified at this step are the fundamental units passed to all further processing steps for analysis and detection of information. The preprocessing process converts the unstructured data into more acceptable data for analysis. On Twitter social network, tweets contain some meaningless and irrelevant words. It is often believed that these words do not affect the tweet's meaning. There are several preprocessing performed in this phase.

Algorithm 3. 2 Preprocessing

Input: F (Tweets CSV File obtained from algorithm 3.1)

Output: Clean_Tweet // Dataset clean

Begin

1. Tweets = Read (F[Text]) // Read only tweet text from file
2. For tw in Tweets
3. Remove all hex characters from string (tw)
4. Remove less 3 char (tw)
5. Remove Emails (tw)
6. Remove distracting single quotes (tw)
7. Remove new line characters (tw)
8. Remove Web http (tw)
9. Remove Stopwords (tw)
10. Tokenize (tw)
11. Check spelling (tw)
12. Converted Lowercase (tw)
13. Lemmatization (tw)
14. End for
15. Return (Clean_Tweet) // Return clean tweet

End

The preprocessing step in Algorithm (3.2) aims to convert the Twitter raw data of CSV file format to readable files required by the next phase of the proposed model. Tweets include irrelevant words, emotions, symbols, numbers, emails, and links. These things have no impact on the text's meaning. Therefore, it is challenging to analyze content from a collection of tweets with these terms. Before any analysis, these terms should be eliminated from tweet to avoid unexpected errors and save

computational resources. Tweets are cleaned by converting the unstructured text into a more understandable structured format for analysis. The preprocessing procedure is started by cleaning meaningless terms, tokenizing, removing stopwords, and lemmatization. The following procedures have been executed as part of the tweet preprocessing step:

- Extracting mentions from tweets that begin with the '@' symbol and follow by another username anywhere in the text because users' names show no meaning in sentences.
- Extracting retweets from tweets that begin with the 'RT' word and follow by '@username' in the tweet.
- Extracting hashtags from tweets that begin with the '#' symbol and follow by hashtag word in the tweet.
- Non-alphabetic characters like punctuation, newline, digits, and "Emoji" are removed from each tweet throughout the cleaning process.
- All words less 3 character, signs, Web URL, and emails removed. They would not give an important meaning in the text analysis.
- Removing stopwords which are English words that don't add much to sentence meaning. They may be ignored entirely without affecting the sentence's meaning, for example, words like a, she, could, and has.
- Splitting the tweets terms into tokens. The tokenization process splits the tweets terms into tokens that can be passed to the next

normalization operation. Spaces are usually used to separate the sentence into tokens.

- Checking the spelling of the tweet's words. An English dictionary was used to complete this phase.
- All tokens are converted into lowercase.
- Checking the lemmatization of tokens by mapping into their origin roots. This process uses the lemmatize, which searches the word database for a lemma of words.
- Part of speech (Pos-tagging) utilizes by parsing the words into nouns, verbs, adjectives, and other parts of speech. Only the nouns were chosen in the case studies.

3.5 Twitter Network Construction

In Twitter, People usually post text according to what they want to share on social media. Then, users interact with post by retweet, reply or like. The interaction among users need to extract from tweets to build interaction networks (i.e., retweet, mention, and hashtag). Interactive Networks are created as a graph representing a set of nodes that connect through edges and assign these edge weights. Nodes represent individuals, or blogs while edges could represent friendship, a hyperlink relationship between two blogs, or any other relationship. A retweet is a Twitter attribute that reposts a tweet displayed by another user using 'RT user name' at the beginning of the tweet. A mention is a Twitter attribute that references another user name any place in the tweet text using '@user name'. The network is a design as *G (V, E, W)* that *V* describes the set of the users (nodes), and *E* represents the set of the link between the user

(edge). *W* is a set of weights values assigned to the edges *E* depending on user interactions. The building network from Twitter is illustrated in Algorithm (3.3).

Algorithm 3. 3 Build Retweet Network, Mention Network, Hashtags

Input: F (Tweets CSV File obtained from algorithm 3.1)

Output: RT, M, H //Retweet and Mention networks, Hashtags

Begin

```
1. Tweets = Read (F[Text])                  // Read only tweet text from file
2. For tw in Tweets
3.     Extract Mentions (tw)                // Name of user mentioned
4.     list of retweet user (U1, U2, W)
5.     Extract Retweets (tw)                // Name of user post retweeted
6.     list of mention user (U1, U2, W)
7.     Extract Hashtag (tw)                 // hashtag write in tweet
8.     List of hashtags
9. End for
10. For users in Mention user list
11.    create edge (U1,U2,W)                // build Mention graph
12. For users in Retweet user list
13.    create edge (U1,U2,W)                // build Retweet graph
14. Return RT, M, H
```

End

Figure (3.3) represents users interact among them based on retweet and mention action. Let's "U1" is a user that mentioned another user "U2" in his post, and the number "2" above the arrow represents the number of times user "U1" mentioned user "U2" in the tweet. Further, Lets "U2" is a user that retweets another user "U3" on his profile page and the number

"3" above the arrow represents the number of times user "U2" retweet user "U3" tweets.

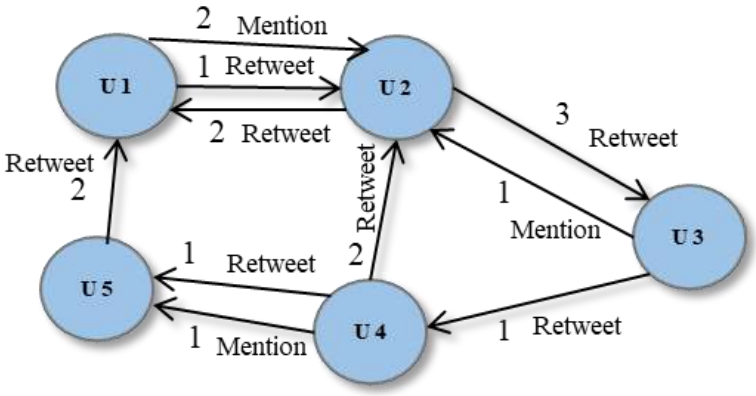


Figure 3. 3 Sample Graph Network

3.6 Topic Model

After the preprocessing step, each tweet was changed into an appropriate format for the next stage - topic model. Topic model is one of the important methods in the present work. Topic model is a process for automatically discovering hidden topics in a documents. Topic algorithms need rich text when detecting topics in documents, while Twitter has difficulty finding topics because it is an informal short text post. It is necessary to use topic model to support the identification of the trends of the tweets. Therefore, the proposed approach has used hashtags terms as a part of the tweet instead of deleting them by enriching tweets. Users use hashtags on tweets to show important subjects or events in the discussion. The enriched tweets would be mapped into a numerical representation of tweets collection called TF-IDF to detect topics. Topics

for tweets can be detected using the proposed method that exploits hashtags to improve the LSA topic modelling performance.

3.6.1 Hashtags Enrichment

Many issues have emerged in the Twitter environment because a tweet is a concise text containing many spelling errors and non-standard grammar. This can result in an extremely low number of words co-occurrence in a set of tweets. Therefore, model will be useless. This issue was addressed by providing a method to exploit hashtags to improve the LSA performance. Hashtags have store important content that can help to discover topics. Hashtag enrichment is overcoming the disadvantages of tweets' short length. In this work, a strategy was developed to overcome the lack of content in tweets. This process aims to enrich the content of each tweet by including more information. This was accomplished by employing both the tweet itself and hashtags terms. The hashtags terms were extracted from the dataset and appended to the cleaned tweets constructing information tweet.

3.6.2 TF-IDF Measurement

TF-IDF is a technique used for calculating the importance of words in a group of tweets. Commonly, each word gives a value to show its importance in the tweet or corpus. The enriched tweets would be mapped into a numerical representation of tweets collection using TF-IDF. The TF-IDF was calculated as Equation (2.3) in chapter 2 using a dictionary with (tweets, token) pair as key and TF-IDF score as the value. It is determined by iterating over all the tweets, which utilize the counter to calculate the frequency for each word in the dataset. TF-IDF score has

calculated the words used in a tweet and their usage compared to terms used in all tweets. For each word, TF shows how many times a word is used in a whole tweet, and IDF shows how important the word is in all tweets.

3.6.3 Latent Semantic Analysis (LSA)

LSA algorithm has been applied on TF-IDF weighted tweets to find the topics. LSA was executed with diverse topics numbers until suitable topics were found for the dataset. LSA is represented tweet as a word-document matrix and utilizes Singular Value Decomposition (SVD) to decompose this matrix to derive k number of topics. In this work, the dataset is defined as a word-tweet matrix. Each row represents a particular word, and the column stands for the tweet. Each cell entries contain the TF-IDF weighting of each word in a tweet. SVD is conducted on the matrix to reduce rank into the k most significant values are included. The SVD algorithm then divides words-tweets matrix into three matrices multiplications as an Equation (2.5) in chapter 2: the first matrix W corresponding to words, the second matrix S expresses topics, and the third matrix D corresponding to tweets. The final picture finds the efficient number of k dimensional in the smallest space to represent data. Each word tweet and word is currently described as a k-dimensional vector in the area originated by the SVD. LSA uses a truncated SVD for its semantic space to deal well with words with the same meanings, as shown in the Algorithm (3.4).

Algorithm 3.4 LSA Topic Model

Input: Clean_Tweet (Cleaned Tweets obtained from algorithm 3.2), H (Hashtags obtained from algorithm 3.3)

Output: T (Topics)

Begin

1. For each Tweet in Clean_Tweet
2. Enriched Tweets = Tweets + H //Enrich Tweets with hashtags
3. For each word in Enriched Tweets
4. Calculate TF-IDF for each word in tweet as Equation (2.3)
5. Dictionary(Enriched Tweets, TF-IDF) //set of words of all tweets
6. k = select no. topics
7. SVD = Truncated SVD (n_components = k) //set SVD with number of topic
8. LSA = SVD (Dictionary) //decompose dictionary into k topics
9. Return (T)

End

3.6.4 Evaluation of LSA Model

This work measured the improved topic model by determining topic coherence. The best number of topics is discovered for all the tweets in the dataset by finding the best coherence value as Equation (2.9) in chapter 2. The LSA method was trained with various topics numbers used to evaluate the average coherence value for each topic. The coherence score estimates the value of the coherence of words in the topics. If the coherence value appears to be improving, it might be better to choose the model with a high coherence value. Then, each tweet in the dataset assigns a topic that reflects the topic user discusses. Each topic consists of words that are ordered according to their probability. Tweets allocated topics

based on the highest probability. This is helpful since it effectively classifies the tweets, sorting them according to the topic they are most likely to belong to them. Each tweet word is compared with the topic word and find the dominant topics for the tweet in the dataset, as shown in the Algorithm (3.5).

Algorithm 3.5 Topics Coherence

Input: (limit, start, step), Clean_Tweet, dictionary (obtained from algorithm 3.2)

Output: k (best number of topic) , Tweets_topics (dominant topic)

Begin

1. For num_topics in range(start, limit, step):
2. Call LSA model(num_topics)
3. Calculate coherence value as in equation (2.9).
4. list = coherence value, num_topics
5. End for
6. For value in list of coherence value
7. Compare coherence value
8. k num_topics = Select high coherence value
9. End for
10. For each Tweet in Clean_Tweet
11. call LSA model (k num_topics)
12. Find similar between topic words and tweet words
13. Tweet_topics = assign dominant topic for each Tweet
14. End for
15. Return (k num_topics, Tweets_topics)

End

3.7 Sentiment Analysis

Sentiment analysis is the process of computationally recognizing and classifying opinions, emotions, and feelings represented in tweets. The main goals of sentiment analysis are to identify and extract emotional information to determine whether the user has a positive, neutral, or negative opinion about a given topic.

3.7.1 Support Vector Machine (SVM)

SVM is a classification model for finding data classes. First, tweets obtained from the topic model are transformed into a vector of numbers. Word vector is the most significant structure during the classification process. The vectorization technique was used for the support vector machine method. To create this valuable representation for our SVM classifier, Bag of Word (BOW) was used with a unigram approach on the tweets dataset, which counts occurrences of the word in the tweet. The vector size is the same as the number of words. All tweets are converted into word vectors to be analyzed by a classifier. Initializing SVM and fitting the training labeled tweets to trained classifiers. SVM is trained by plotting the tweets data (words vector) into space as a point then initializing hyper parameter to detect the classification. The training tweets help the SVM convert the inputs into dimensional feature space. It uses margin values and support vectors to calculate distance among points as Equation (2.6). SVM has identified an isolated hyperplane that maximizes the margin among distinct classes. The SVM classifiers trained with labeled tweets are used to predict the unlabeled tweets. The SVM is labeled tweets as positive (1), neutral (0), and negative (-1).

Algorithm 3.6 SVM Method

Input: Tweets_topics (unlabeled tweet obtained from algorithm 3.5), T(Training dataset), L (labels)

Output: Label_Tweets // Tweet classification

Begin

1. Trained = read (T) // Load Training tweets
2. L = {label 1, label 0, label -1} // Label classes
3. For each Tweet in Trained
4. BOW = bag of word(Tweet) //Convert label tweets into BOW Vector
5. SVM_ Train = SVM (BOW, L) //Trained SVM with label data
6. For each Unlabeled _Tweet in Tweet_topic
7. Unlabeled _BOW = bag of word(Unlabeled _Tweet) //Unlabeled tweets convert to BOW
8. SVM_ pred= SVM_ Train (Unlabeled _BOW) //SVM classify unlabeled tweets
9. Label_Tweets = SVM_ pred //Assign label to the Tweets
10. Return (Label_Tweets)

End

3.7.2 Evaluation of SVM

In predictive analysis, classification evaluation determines what types of data are classified as true and what kinds of data are labeled as false by the classification model adopted. The most often used metrics for evaluating support vector machine are accuracy, precision, and recall. Usually, four primary parameters are used to assess SVM performance (true positive, false negative, false positive, and true negative) illustrated as follows:

- True Positive Label (TPL): It occurs when SVM correctly label the tweets as a positive opinion.
- False Negative Label (FNL): It occurs when SVM incorrect label the tweets as a negative opinion, but it is actual positive opinion.
- False Positive Label (FPL): It occurs when SVM incorrect label the tweets as a positive opinion, but it is actual negative opinion.
- True Negative Label (TNL): It occurs when SVM correctly label the tweets as a negative opinion.

These parameters are used to calculate accuracy as Equation (2.15) in chapter 2. It is the proportion of the total number of correct predictions.

3.8 Community Detection

On Twitter, the networks might be harder to get information about communities. Most existing algorithms were designed to detect community using structure networks (nodes and edges) that do not represent user data into the relationship. Community detection methods could have an immediate influence on network organization and function. Including information into the nodes and community can help discover the perfect community. These community nodes can contain information about users' opinions about specific topics or what topics users discuss. Therefore, enhancing and developing the community detection algorithm was the primary purpose of this work. The proposed model tries to find communities by utilizing the attributes of tweets in which the node within has more information about user connections such as (topic and opinion).

It adds more information on communities rather than linking relationships between nodes. The strategy combines the multiple attributes of the user's tweet with the link of the network topology. Community detection is a process of grouping similar users into the same cluster that has tried to discover optimal groups of users. The attributes are represented using the retweet and mention actions, and the tweets are analyzed based on the topic model and sentiment analysis. These techniques merge to get a new weighted network to enhance the quality function of modularity and a Constant Potts Model (CPM) for Leiden community detection.

3.8.1 Enriching Community Network

The Twitter network consists of nodes that link and interact with users or organizations by edges. This structure network has many attributes within the network topology. Mathematically, the new network for the proposed model is described as a weighted network *(V, E, W). V* denotes a collection of vertices (nodes) $\{v_1, v_2, \dots, v_n\}$, *E* denotes a collection of edges $\{e_1, e_2, \dots, e_n\}$, and *W* denotes a collection of weights values. The new network creation strategy employs topics and sentiment with retweet and mention networks that are a built-in preprocessing step. The presented method assists in updating edges weights between nodes used by the community detection method to discover communities on the network. Unlike the structure network that only has link relation, this new network has a new weight that combines information about topics and sentiment that help to improve Leiden community detection in the Twitter network.

Weights are assigned to the edges between users based on the type of network interaction. The retweet network weights are determined by how many people have reposted another user's tweet. Similarly, the mention network weights are determined by the number of times a user was mentioned in another user's tweet body. At this stage, the results obtained from LSA and SVM with mention and retweet network in the previous step are exploited to calculate new weights edges among users. First, RT and M obtain were constructed the retweet and mention networks that contain weight reflect the interaction among users. Second, Tweets_topics were obtained from LSA topic model, representing the topic discussed by the user in the tweet. Label_Tweets were obtained from SVM, which represent the opinion of users. As illustrated below, a new weight is calculated by adding the normalized weight of user retweet and mention plus topic and sentiment if they share a similar subject or opinion.

$$w_{u_1,u_2} = (Norm\_w(RT) + Norm\_w(M)\ ) + T + S) \qquad (3.1)$$

$$Norm\_w(RT) = \left(\frac{Rt\ u_1,u_2}{Rt}\right) \times U \qquad (3.2)$$

$$Norm\_w(M) = \left(\frac{Mt\ u_1,u_2}{Mt}\right) \times U \qquad (3.3)$$

Where $Rt\ u_1, u_2$ describes the number of retweets between two users. U and RT are the total numbers of users and retweets, respectively. $Mt\ u_1, u_2$ denotes the number of mentions between two users. Mt describes the total number of mentions.

Algorithm 3.7 New Weighted Network Calculation

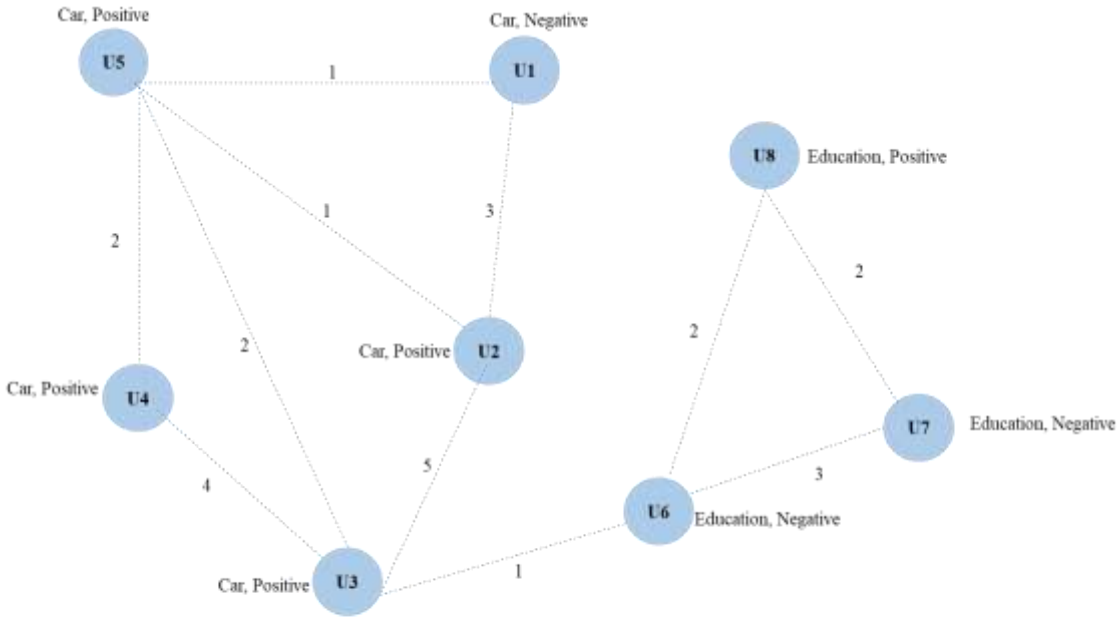
Input: RT, M, Tweet_topics, Label_Tweets

Output: New _ network // New weights Network

Begin

1. For raw in range(RT) // Retweet network
2. T, S = 0 // Initialize variables
3. Norm _w(RT)= Normalize (raw [weight]) //Normalization retweet weight As Equation (3.2)
4. For raw in range(M) // Mention network
5. If Structure Similarity between RT and MT //Share same edges
6. Norm_w(M)= Normalize (raw [weight])//Normalization Mention weight As Equation (3.3)
7. If users have similar topic (Tweet_topics) //Users share same topic
8. T = 1
9. If users have Similarity sentiment (Label_Tweets) //Users share same opinion
10. S = 1
11. Weight calculated as Equation (3.1)
12. End for
13. Return (New _ network)
14. End

A new edge network is calculation on the model, as shown in the Algorithm (3.7). The weight of edges between users of the retweet network is normalized by the total number of retweets and multiple by the total number of users. Also, the weight of edges between users of the mention network is normalized by the total number of mentions and multiple by the total number of users. The retweet and mention network weights rely on the number of times the user retweeted and mentioned another user's tweet.



Nodes	Links	Weight	Topic	Sentiment
U1	U2, U5	3 , 1	Car	Negative
U2	U1, U3, U5	3 ,5 ,1	Car	Positive
U3	U2, U4, U5, U6	5,4,2,1	Car	Positive
U4	U3, U5	4, 2	Car	Positive
U5	U1, U2, U3, U4	1,1,2, 2	Car	Positive
U6	U3, U7, U8	1, 3, 2	Education	Negative
U7	U6, U8	3, 2	Education	Negative
U8	U6, U7	2, 2	Education	Positive

Figure 3. 4 New Weight Network

Weight of retweet edges update based on Twitter attributes by three procedures. First, suppose one user mentioned other users in tweets. In that case, the edge weight between users is increased by the normalized

value of the mentioned weight. Second, If two users share a similar topic, the edge weight between users raises by one. Third, If two users share a similar opinion, the edge weight between users raises by one. The new network consists of information about nodes, edges, weight, topics, sentiment, as shown in Figure (3.4). Each edge assigned new weight calculated from combined retweet network with mention and tweet attributes. Each node (user) contains information about the topic discussed and his sentiments. This new weight network will be used in Leiden community detection to discover users communities.

3.8.2 Enhancing Leiden Community Detection

The enhancing Leiden community detection is started by obtaining the new weight among the nodes. The Leiden algorithm is generally seen as one of the most useful algorithms for detecting communities. The Leiden algorithm guarantees communities are well-connected. It concentrates on a partition where all subsets of all communities are assigned locally optimally. It is faster method because it relies on a quick local move routine. The enhancing Leiden algorithm consists of these steps:

1. Start by calculating new weights using Algorithm (3.7), combining topics and sentiment with retweets and mentions.
2. Then creating a singleton node community.
3. Nodes exploit the new weights to find the best partition that nodes are moved locally from one community to another.
4. The refinement is fine-tuned partition by the node can be split or combined communities in which the quality function increases.

5. Network aggregation based on this refined partition. The aggregate network's first partition is created using the non-refined partition.
6. The local moving, refining, and aggregation procedures are repeated until no further improvements are possible.

Algorithm 3.8 Enhancing Leiden Community Detection

Input: RT, M (obtained from algorithm 3.3), Tweet_topics (dominant topic) (obtained from algorithm 3.4), Label_Tweets (Tweet classification) (obtained from algorithm 3.6)

Output: P Communities Partition

Begin

1. Call New Weighted Network (RT, M, Tweet_topics, Tweet_topics)
2. **Do**
3. P ← Fast _Local _Move(G, P, N_ weights) //Move nodes between communities
4. P == N (G) //Every community is one Node
5. **If** not done **then**
6. p_{ref} ←Refine_Partition(G, P) //Refine partition P
7. G ←Aggregate_Graph_refined(G, p_{ref}) //Create aggregate graph
8. P ← Partition(V(G)) // Maintain partition P
9. End if
10. **While** not done
11. Return flat *(P) //Communities

End

3.8.3 Evaluation of Community Detection

After detecting communities for the Twitter networks, it is helpful to assess the detected communities to answer the question of the enhancement. Two methods performed the evaluation. The current model measured the improved Leiden algorithm by determining modularity and a Constant Potts Model (CPM). Modularity is often used in optimization approaches for discovering community structure in networks. Modularity is a network structural metric that evaluates how well a network can be divided into groups. The high modularity value refers to more connections among nodes inside groups but fewer connections among nodes in different groups. Modularity was calculated as Equation (2.16) in chapter 2.

The constant Potts Model (CPM) is a network metric that evaluates how a network is divided into communities. High CPM value has the best communities partition dense links between nodes inside communities. CPM was calculated as Equation (2.17) in chapter 2.

Chapter Four

Experimental Results and Discussions

4.1 Overview

This chapter introduces and clarifies the implementation and experimental results of the proposed model, which has been performed on Twitter datasets. At first, the Twitter API is utilized to collect data. Next, Preprocessing is applied on Twitter data. After that, the topics and sentiment of users' tweets were discovered. Consequently, the new weights of the network are calculated by combining the structure network with the content of tweets. At last, but not least, the chapter will investigate the Leiden community detection results after calculation the new weight. Finally, the improved Leiden results were compared with other methods on the Twitter dataset.

4.2 System Requirements

- Hardware, Processor: Intel(R) Core(TM) i7-8565 CPU @ 2 GHz
- Memory: 20 GB RAM.
- Operating System: Windows 10 Pro 64-bit.
- Programming Language: Python 3.7
- IDE Environment: PyCharm 2019.1

4.3 Twitter Datasets

Twitter is one of the most widely used social media networks worldwide. Existing Twitter datasets couldn't be employed for this proposed model because they have only text tweets or structural relationships among users. In this work, the model required a dataset that can offer a structured network and the users' texts both, not just the information about who is linked to whom. Moreover, retweets, mentions,

and hashtags are also needed for those tweets. Therefore, real tweet data have been collected in this work from Twitter using the API interface. Twitter provides a search API for developers to get recent real tweets published daily. The Twitter search API is used to collect tweets using keywords and hashtags. Tweets come from users who actively speak about a particular subject.

In the current work, the two raw datasets were collected from Twitter using APIs as stated in (3.1) algorithm. These raw data will be preprocessed by cleaning and putting them in order in the dataset structure to be used in the future for studies related to this aspect. The first raw data collected was 10406 tweets regarding the social networking website, phone, and electronic devices. The second raw data collected was 30910 tweets regarding anxiety, education, cars, and entertainment. These tweets were collected between the years 2019-2020. The downloaded data contain a set of tweets, mentions, hashtags, retweets, and users' details. The number of gathered tweets based on hashtags and keywords is stated in Table (4.1).

Table 4. 1. Details of Raw Data Collected

	Search Query	**No. of Tweets**
Raw Data 1	Samsung, phone, Twitter, android, IOS, apple #Facebook, #Twitter, #IPhone	10406
Raw Data 2	anxiety, stress, school, car, mobile, cinema # anxiety, #movie, #vacation, # education	30910

4.3.1 Data Extraction

Tweets were gathered from Twitter and stored in JSON format. Figure (4.1) shows a sample of a JSON Format. On Twitter, a tweet has a set of attributes more than 150 attributes. Some attributes were extracted from tweets to use in this work such as: Created at, ID, Text, and User name see Table (4.2). A CSV spreadsheet with the prior data along with a column was built. JSON data were converted into key-value pairs to CSV. For datasets collected from Twitter using APIs, attributes are extracts from tweets and stored in CSV files.


```
"Created_at": ": "Sun May 25 20:11:36 +0000 2020",
          "id": 1103654215276136,
         "text": " beating you @heralteric @nust. I sense the latter days of Xiaomi
                  begin from phone #apple "
          "source": "<a href=\"http://twitter.com/download/iphone\"
          "user": {"id": 11259386409740224,
                    "name": "Bah Fulerton",
                    "screen_name": "BahFulerton2",
                    "location": "Arkansas, USA", …}
           "entities": {"hashtags": [{"text": " apple ", "indices": [91, 96]}], "urls": [],
           "user mentions": [{"screen_name": " heralteric, ", "name": "Charles
            Dave", "id": 3236653, "indices": [14, 30]}
```


Figure 4. 1 Simple JSON Format Tweets

Table 4. 2 Sample of CSV Tweets

	Created_at	ID	Text	User name
1	2020-04-24 10:04:30	2186261	I'm a recent @Samsung user, a short bit frustrated with it! must I change to #IPhone @jack @johnsa	User1
2	2020-04-24 10:11:23	1183656	@Sam When I have nothing to do , when I'm doing homework, I'm on twitter I don't do ANYTHING	User2
3	2020-04-24 10:14:55	2085456	Galaxy Vs iPhone: Which smartphone will wins? I ask @phahk to test and get result galaxy apple #Android	User3
4	2020-04-24 10:17:45	1193275	I remember when I opened #twitter the first follow I gave was to @Istook he is my best Friend	User4
5	2020-04-24 10:18:20	2298252	open Facebook because you do not entertain your people so that we no longer go away so many plebs Â¬Â¬	User5
6	2020-04-24 10:24:29	1296553	beating you @heralteric @nust. I sense the latter days of Xiaomi begin from phone #apple.	User6
7	2020-04-24 10:24:30	1525357	The most unnatural things Siri has spoken so far. I am so happy to tell me this word that apple provided Siri a mood! http://t.co/ehgpp via @Hlace	User7
8	2020-04-24 10:24:38	2426324	Thanks to #Samsung for exchanging my phone screen during my shopping today, available of guarantee.	User8
9	2020-04-24 10:24:40	1093242	When I said I'm trying to sleep, I actually mean I'm supposed to appear for 30 minutes on Twitter before bedtime! <3	User9
10	2020-04-24 10:24:42	2178643	I'm disordered up because each time I write a tweet text I feel worried since I have a few characters! It's becoming difficult to follow all of the text	User10

4.4 Use Case Study

Many experiments to assess the proposed work have been executed on Twitter raw data collected. The proposal model was applied on the collected Twitter data of 10406 Tweets.

4.4.1 Preprocessing Stage

While collecting data from Twitter faced a few challenges. Several tweets contain no text such as pictures and videos. Many users write in other languages in tweets described in English. Some tweets were advertisements of events or businesses company. Few tweets didn't include any text, only a user name. Few tweets have the hashtagged term on the tweet. Many people had duplicated others' tweets by retweets.

Therefore, the preprocessing techniques were implemented to clean the collected data and build the dataset. It is one of the most critical aspects of this work because tweets contain different of terms, phrases, hashtags, mentions, and URLs. The data cleaning procedure followed these actions. The tweets were checked for duplication and deleted consecutively. We extract mentioned names (i.e., @John) and retweet names (i.e., Rt @Malik) in tweets. All hashtags were extracted from the tweets (only the # symbol were removed). These mentions and retweets are used in the next phase of the current model to build networks. Hashtags were also used in the next step to enrich tweets before discovering topics. All words less 3 char, signs were removed. Applying spell check that check and correct words (i.e.,'happpy' became 'happy'). Removing non-alphanumeric terms (not in a-z, A-Z, 0-9). All URLs and emails were removed (i.e., https://t.co/VzS5ahNPt, ha@gmail.com). Tweets words are

transformed to lower case to remove ambiguity and standardization. Stopwords are a group of regularly used words in the English language, (i.e. 'I', 'a', and 'the'). These words have been deleted since they are meaningless and increase the number of words without any benefit. The lemmatization process converted words that contain prefixes and suffixes by mapping their origin roots (i.e. running and ran are converted to run). POS tagger is to assign grammatical information to tokens (i.e. everything is all about life' are mapping to ('everything', 'NN'), ('is', 'VBZ'), ('all', 'DT'),('about', 'IN'), ('life', 'NN'), ('.', '.')). A sample of preprocessing operations on dataset have shown in Table 4.3.

Table 4. 3. Sample of Preprocessing Operation

Processing Operation	Result Action
Text	@bery & @acook are really about to cause me to throw my phone in the garbage and buy an #Apple? New device!
Extract Mention from Tweet	@bery , @acook
Extract Hashtag from Tweet	#Apple
Non-alphabetic characters	are really about to cause me to throw my phone in the garbage and buy an New device
Remove Stopwords	really cause throw phone garbage buy New device
Splits the tweets into tokens	[' really ', ' cause ', ' throw ', ' phone ', ' garbage ', ' buy ', ' New ', ' device ']
Check the spelling	[' really ', ' cause ', ' throw ', ' phone ', ' garbage ', ' buy ', ' New ', ' device ']
Convert to Lowercase	[' really ', ' cause ', ' throw ', ' phone ', ' garbage ', ' buy ', ' new ', ' device ']
Check the lemmatization	[' really ', ' cause ', ' throw ', ' phone ', ' garbage ', ' buy ', ' new ', ' device ']
word Part of speech	[('really', 'RB'), (' cause', 'VB'), ('throw', 'VB'), ('phone', 'NN'), ('garbage', 'NN'), ('buy', 'VB'), ('New', 'NNP'), ('device', 'NN')]

Table 4. 4. Example of Tweets in Dataset

	Tweets
1	I'm a recent @Samsung user, a short bit frustrated with it! must I change to #IPhone @jack @johnsa
2	@Sam When I have nothing to do , when I'm doing homework, I'm on twitter I don't do ANYTHING
3	Galaxy Vs iPhone: Which smartphone will wins? I ask @phahk to test and get result galaxy apple #Android
4	I remember when I opened #twitter the first follow I gave was to @Istook he is my best Friend
5	open Facebook because you do not entertain your people so that we no longer go away so many plebs Â¬Â¬
6	beating you @heralteric @nust. I sense the latter days of Xiaomi begin from phone #apple.
7	The most unnatural things Siri has spoken so far. I am so happy to tell me this word that apple provided Siri a mood! http://t.co/ehgpp via @Hlace
8	Thanks to #Samsung for exchanging my phone screen during my shopping today, available of guarantee.
9	When I said I'm trying to sleep, I actually mean I'm supposed to appear for 30 minutes on Twitter before bedtime! #Twitter <3
10	I'm disordered up because each time I write a tweet text I feel worried since I have a few characters! It's becoming difficult to follow all of the text

The preprocessing step in Algorithm (3.2) was applied on tweet dataset to change the tweets to the readable format required by the next phase of the proposed model. Irrelevant words, emotions, symbols, numbers, emails, and URLs that appear in tweets that are unlikely to help text mining, such as prepositions, articles, and pronouns, see Table (4.4) and (4.5).

Table 4. 5. Result of Preprocessing on Tweets

	Cleaned Tweets
1	[' recent ', ' user ', ' short ', ' bit ', ' frustrated ', ' change ']
2	[' homework ', ' twitter ']
3	['galaxy', 'iphone', 'smartphone', 'wins', 'ask', 'test', 'get', 'result', 'galaxy', 'apple']
4	[' remember ', ' opened ', ' follow ', ' gave ', ' best ', ' Friend ']
5	[' open ', ' Facebook ', ' entertain ', ' people ', ' longer ', ' go ', ' away ', ' plebs ']
6	[' beating ', ' sense ', ' latter ', ' days ', ' Xiaomi ', ' begin ', ' phone ']
7	[' unnatural ', ' things ', ' Siri ', ' spoken ', ' far ', ' happy ', ' tell ', ' word ', ' apple ', ' provided ', ' Siri ', ' mood ']
8	[' exchanging ', ' phone ', ' screen ', ' shopping ', ' today ', ' guarantee ']
9	[' said ', ' trying ', ' sleep ', ' mean ', ' supposed ', ' appear ', ' minutes ', ' Twitter ', ' bedtime ']
10	[' disordered ', ' time ', ' write ', ' tweet ', ' text ', ' feel ', ' worried ', ' characters ', ' difficult ' , ' follow ', ' text ']

After cleaning tweets, the part of speech or POS-tagging is performed where every word in the tweet refers to the role the word plays in a sentence. i.e., noun, verb, adjective...etc. The POS tagging applies to selecting the nouns only to discover topics on tweets in the next phase. Table (4.6) shows all parts of speech words in the tweets dataset.

Table 4. 6. Result of POS-Tagging on Tweets

	POS-Tagging
1	[('recent', 'JJ'), ('user', 'NN'), ('short', 'JJ'), ('bit', 'NN'), ('frustrated', 'JJ'), ('change', 'VBP')]
2	[('homework', 'NN'), ('twitter', 'NN')]
3	[('galaxy', 'NNP'), ('iPhone', 'NN'), ('smartphone', 'NN'), ('wins', 'VB'), ('ask', 'VBP'), ('test', 'VB'), ('get', 'VB'), ('result', 'NN'), ('galaxy', 'NN'), ('apple', 'NN')]

4	[('remember', 'VBP'), ('opened', 'VBD'), ('follow', 'NN'), ('gave', 'VBD'), ('was', 'VBD'), ('best', 'JJS'), ('Friend', 'NN')]
5	[('open', 'JJ'), ('Facebook', 'NNP'), ('entertain', 'VB'), ('people', 'NNS') , ('longer', 'RBR'), ('go', 'VB'), ('away', 'RB'), ('plebs', 'NNS')]
6	[('beating', 'VBG'), ('sense', 'VBP'), ('latter', 'JJ'), ('days', 'NNS'), ('Xiaomi', 'NNP'), ('begin', 'VBP'), ('phone', 'NN')]
7	[('unnatural', 'JJ'), ('things', 'NNS'), ('Siri', 'NNP'), ('spoken', 'VBN'), ('far', 'RB'), ('happy', 'JJ'), ('tell', 'VB'), ('word', 'NN'), ('apple', 'NN'), ('provided', 'VBD'), ('Siri', 'NNP'), ('mood', 'NN')]
8	[('exchanging', 'VBG'), ('phone', 'NN'), ('screen', 'NN'), ('shopping', 'NN'), ('today', 'NN'), ('guarantee', 'NN')]
9	[('said', 'VBD'), ('trying', 'VBG'), ('sleep', 'VB'), ('mean', 'VBP'), ('supposed', 'VBN'), ('appear', 'VB'), ('minutes', 'NNS'), ('Twitter', 'NNP'), ('bedtime', 'NN')]
10	[('disordered', 'VBN'), ('time', 'NN'), ('write', 'VBP'), ('tweet', 'NN'), ('text', 'NN'), ('feel', 'VBP'), ('worried', 'JJ'), ('characters', 'NNS'), ('difficult', 'JJ'), ('follow', 'VB'), ('text', 'NN')]

4.4.2 User Network Formation

Networks are built-in Twitter as an interactive graph representing a group of nodes connected by edges with weights assigned. The mentions and retweets extracted in preprocessing stage are used to build networks. The retweet network *(V, E, W)* is built as in Figure (4.2) and Table (4.7), *V* is a set of nodes in which each node reflects the user name. *E* is a set of edges $\{e_1, e_2, \dots, e_n\}$ representing a retweet relation among users (user name of tweet). *W* a set of weights associated with each edge represents the number of retweets among users. The Mention network *(V, E, W)* is built in Figure (4.3) and Table (4.8), *V* is a set of nodes, each node reflects the user name that mention other user. *E* is a set of edges $\{e_1, e_2, \dots, e_n\}$, represents a mention relation among users. *W* a set of weights associated with each edge represents a number of mentions between users.

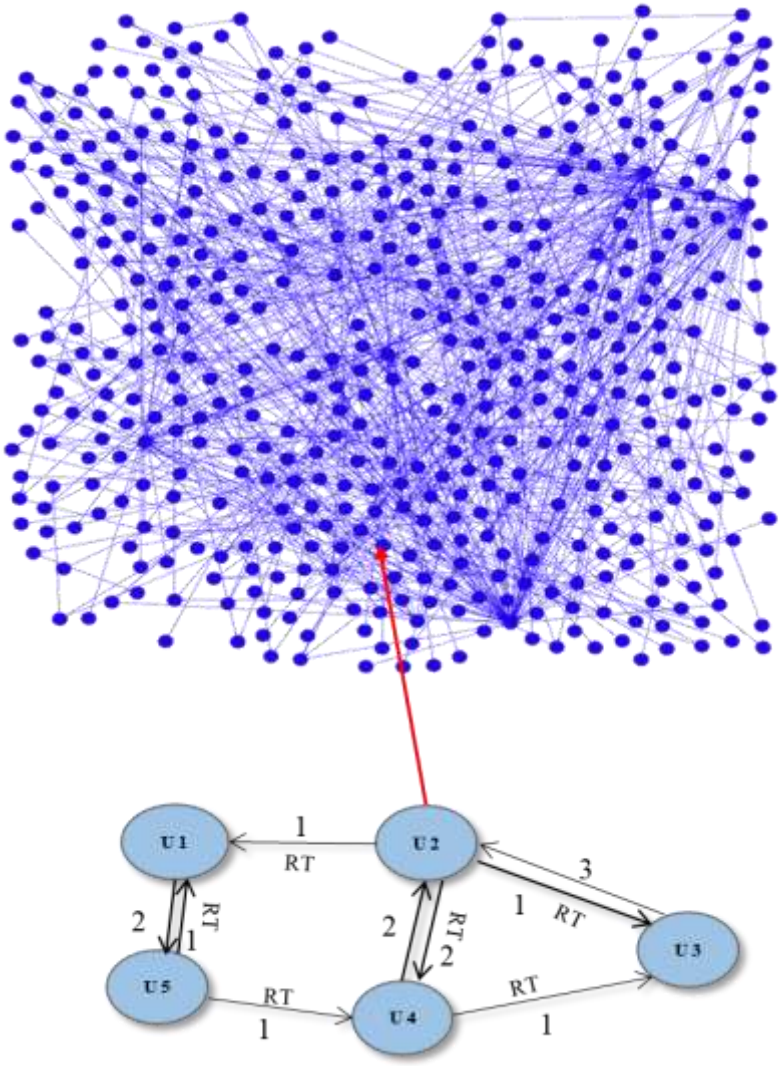


Figure 4. 2 Sample Retweet Network

Table 4. 7. Retweets' Weight in the Network

Nodes	U 1	U 2	U 3	U 4	U 5
U 1	0	0	0	0	2
U 2	1	0	1	2	0
U 3	0	3	0	0	0
U 4	0	2	1	0	0
U 5	2	0	0	1	0

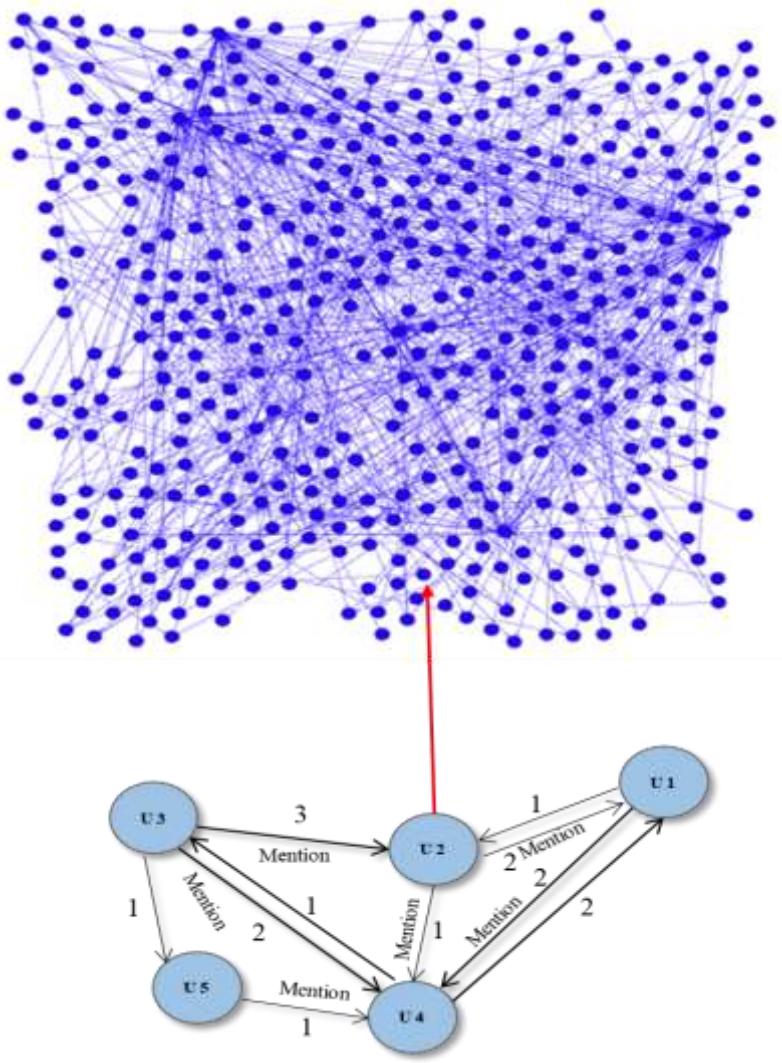


Figure 4. 3 Sample Mention Network

Table 4. 8. Mention s' Weight in the Network

Nodes	U 1	U 2	U 3	U 4	U 5
U 1	0	1	0	1	0
U 2	1	0	0	1	0
U 3	0	3	0	2	1
U 4	2	0	2	0	1
U 5	0	0	0	1	0

4.4.3 User Topic Discovery

After the preprocessing stage, the hashtags enrichment process is applied to cleaned tweets by appending hashtags words extracted from the dataset into tweets. Hashtags words have stored significant information that helps to discover topics. Hashtag's enrichment adds more words that exploited to overcome tweets shortage length for improving the capacity of tweets' contextual meaning. Not all tweets are enriched with the hashtag because not all posts have hashtags depending on the writing manner user.

After the hashtags are appending, the enriched tweets will be mapped into TF-IDF. It is a numerical representation of the tweets dataset used to find topics in LSA. Each tweet in the dataset was converted as a (words, weights), which detects the word's importance in the tweet dependent on value. Figure 4.4 represents the follow of LSA process .

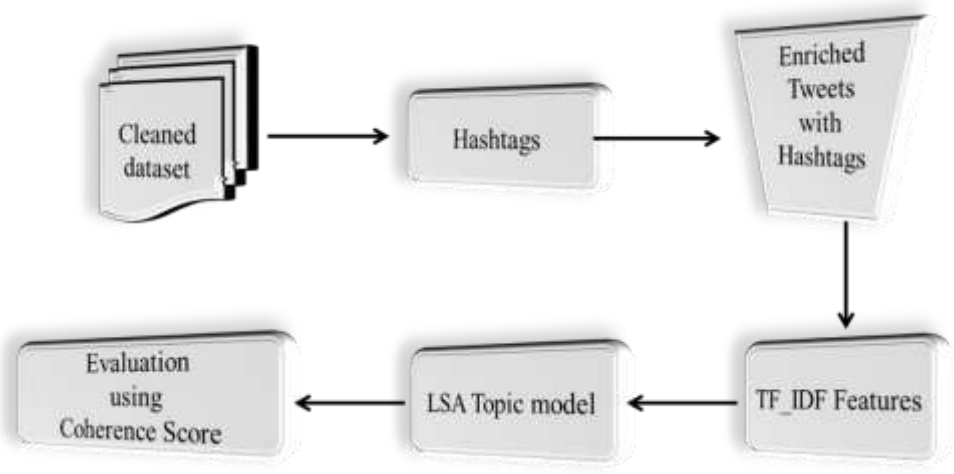


Figure 4. 4 LSA Topic Model Process

Table 4. 9. LSA Coherence Value Comparison

Topic No.	LSA with Hashtags	LSA without Hashtags
2	**0.5737**	**0.5369**
4	0.5551	0.515
6	0.5342	0.4978
8	0.5578	0.5077
10	0.5252	0.4881
12	0.5379	0.508
14	0.5537	0.5136
16	0.5459	0.5099
18	0.5544	0.5341

The LSA topic model technique was used twice in current experiments. The first experiment was used raw tweets without hashtags enrichment to detect the hidden topics. The second experiment was tested on enriched tweets. Several experiments have been executed on various setups to extract latent topics by applying the LSA method in Algorithm (3.4). The experiments used k's topics between 2 and 20 to find the best cluster topics. The coherence score is used to evaluate each experiment of topic modelling. Table (4.9) has shown the results of LSA model experiments. LSA with hashtags enrichment tweets gets higher coherence value than without hashtags enrichment. Using hashtags with tweets helps to improve the coherence score value in LSA result. The experiments showed the best result for LSA with hashtags enrichment tweets when k = 2 with a coherence score of 0.5725 see Figure (4.5), while The best number of topics for LSA without hashtags is k = 2 with a coherence score

of 0.5369. A higher coherence score indicates to a topic is easier to understand the word distribution about what subjects would belong to. The result shows that LSA incorporates hashtags that perform better than LSA without hashtags on most k topic numbers. Using hashtag attribute to enriched tweets helps to improve the coherence score value in our LSA result from 0.5369 in link relationship to 0.5737.

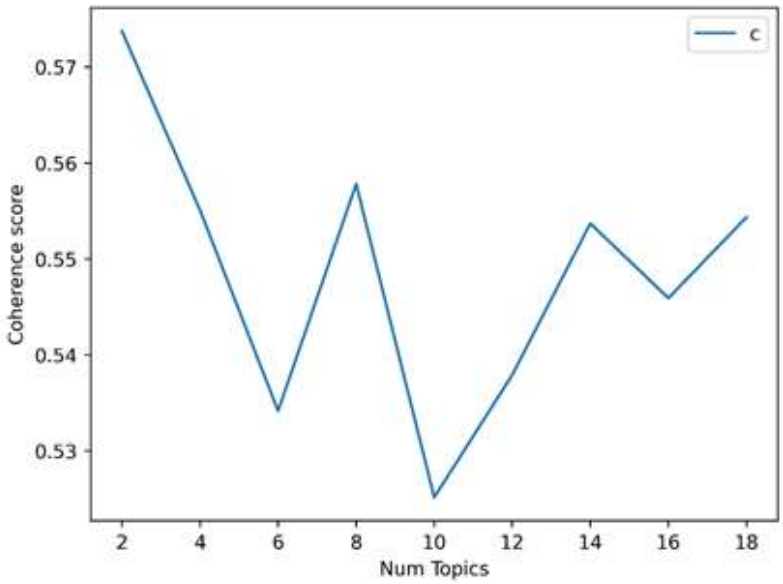


Figure 4. 5 Coherence Score of LSA for Different K

Table (4.10) shows the topics of the best coherence value found after testing LSA with enriched tweets on the different numbers of k. Two topics and their words are discovered by the LSA method on enriched tweets. The words in the topics were ordered according to their probability. Figure (4.6) is a word cloud that plots words when each word is scaled by its probability for an additional indication. For example, the most important words for topic 0 are "twitter" followed by "night",

"tweet", and so on, while the most important words for topic 1 are "phone" followed by, "store", "window", and so on.

All of the words were grouped to build topics. While looking at the word sequence, a topic will be created for each group of words. Thus, words in topic 0 refer to the topic related to social media, while words in topic 1 refer to the topic related to phones or mobile.

Table 4. 10. LSA Topics Result with Enriched Tweets

Topic #0	**Topic #1**
0.996 *twitter	0.963 *phone
0.035 *night	0.101 *store
0.031 *tweet	0.097 *window
0.031 *follower	0.088 *screen
0.026 *day	0.07 * apple
0.019 *time	0.069 *device
0.018 *people	0.068 *week
0.016 *today	0.063 *user
0.014 *sleep	0.059*android
0.013 *name	0.058*mango
0.013 *friend	0.048*iphone
0.012 *life	0.038*purchase
0.01 *Facebook	0.03*update

Figure 4. 6 LSA Word Cloud for Topics on Enriched Tweets

After finding the optimal number of topics using the LSA method, each tweet in the dataset consists of multiple topics. But, usually, only one of the topics is dominant. Each tweet consists of a set of words that represent different topics. Therefore, we need to decide which topic represents each tweet by calculating which topic has the highest contribution and assigning it. Table (4.11) shows assigning the tweet to the topic that has the highest weight in that tweet.

Table 4. 11. Dominant Topics on Tweets

	ID	Tweets	Dominant Topic
1	2186261	I'm a recent @Samsung user, a short bit frustrated with it! must I change to #IPhone @jack @johnsa	1
2	1183656	@Sam When I have nothing to do , when I'm doing homework, I'm on twitter I don't do ANYTHING	0

3	2085456	Galaxy Vs iPhone: Which smartphone will wins? I ask @phahk to test and get result galaxy apple #Android	1
4	1193275	I remember when I opened #twitter the first follow I gave was to @Istook he is my best Friend	0
5	2298252	open Facebook because you do not entertain your people so that we no longer go away so many plebs Â¬Â¬	0
6	1296553	beating you @heralteric @nust. I sense the latter days of Xiaomi begin from phone #apple.	1
7	1525357	The most unnatural things Siri has spoken so far. I am so happy to tell me this word that apple provided Siri a mood! http://t.co/ehgpp via @Hlace	1
8	2426324	Thanks to #Samsung for exchanging my phone screen during my shopping today, available of guarantee.	1
9	1093242	When I said I'm trying to sleep, I actually mean I'm supposed to appear for 30 minutes on Twitter before bedtime! #Twitter <3	0
10	2178643	I'm disordered up because each time I write a tweet text I feel worried since I have a few characters! It's becoming difficult to follow all of the text	0

4.4.4 User Opinion Classification

After obtaining the topic for each tweet in the previous step, the SVM method is applied to recognize and extract the opinion information of users. Before the tweets dataset is classified, we have annotated the tweets data with the help of expert annotators. Annotation is a technique of adding a label to the data to train machine learning algorithms to classify tweets. In this work, each tweet must be labeled as 0, -1, or 1. The label indicates if the tweet is related to a positive feeling, it was given a label 1,

and if the tweet is related to a negative feeling, it was given a label -1. Otherwise, it was given a label 0. Table (4.12) is a sample of the tweets data label.

After annotations, each tweet was transformed into word vectors used in the SVM method. The bag of words is built using a unique word derived from the training dataset. The number of words defines the vector size. The labeled tweets are used to train SVM classifier. The SVM parameters were adjusted on the training tweets. SVM has classified users opinions about a particular topic into positive, neutral, or negative. SVM is used with the adjusted parameter to classify the rest of the unlabeled dataset. The accuracy value on the unlabeled dataset showed that the support vector machine (SVM) had a high performance with an accuracy score of 75% lower than other research results because the data here is not fully classified but rather partly labeled.

Table 4. 12. Labeled Tweets

LABELS	EXAMPLES
Negative	I'm a recent @Samsung user, a short bit frustrated with it! must I change to #IPhone @jack @johnsa
Neutral	I remember when I opened #twitter the first follow I gave was to @Istook he is my best Friend
Positive	The most unnatural things Siri has spoken so far. I am so happy to tell me this word that apple provided Siri a mood! http://t.co/ehgpp via @Hlace

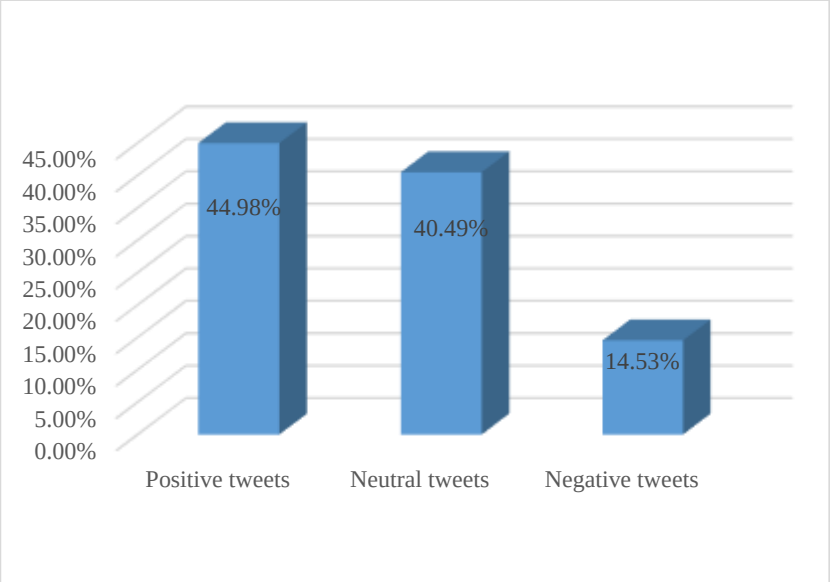


Figure 4. 7 SVM Sentiment Result

The experiments have been executed on tweets dataset to discover user opinion by applying the SVM method in Algorithm (3.6). The result of SVM in Figure (4.7) has shown very few users have negative opinions, while most users have a positive and neutral opinion about two topics. After tweets classification, we can know the user's topic and his feelings about this topic, as shown in Table (4.13).

Table 4. 13. Opinion of Users Tweets

ID	Tweets	Dominant Topic	Sentiment
2186261	I'm a recent @Samsung user, a short bit frustrated with it! must I change to #IPhone @jack @johnsa	1	-1
1183656	@Sam When I have nothing to do , when I'm doing homework, I'm on twitter I don't do ANYTHING	0	0

2085456	Galaxy Vs iPhone: Which smartphone will wins? I ask @phahk to test and get result galaxy apple #Android	1	0
1193275	I remember when I opened #twitter the first follow I gave was to @Istook he is my best Friend	0	0
2298252	open Facebook because you do not entertain your people so that we no longer go away so many plebs Â¬Â¬	0	0
1296553	beating you @heralteric @nust. I sense the latter days of Xiaomi begin from phone #apple.	1	-1
1525357	The most unnatural things Siri has spoken so far. I am so happy to tell me this word that apple provided Siri a mood! http://t.co/ehgpp via @Hlace	1	1
2426324	Thanks to #Samsung for exchanging my phone screen during my shopping today, available of guarantee.	1	1
1093242	When I said I'm trying to sleep, I actually mean I'm supposed to appear for 30 minutes on Twitter before bedtime! #Twitter <3	0	0
2178643	I'm disordered up because each time I write a tweet text I feel worried since I have a few characters! It's becoming difficult to follow all of the text	0	-1

4.4.5 Enhancing Community Detection

Detecting the community is an essential process when analyzing social networks. In this model, the new weight edge was calculated first to combine user content and link relation. For each edge, there should be a weight serving as an interaction degree of a node upon the destination node, or it can be interpreted as a strength of the relationships among the nodes. The weights must be calculated first to get the community detection process done. The key point in this step was to calculate a new

weight that incorporates the topic and sentiment of users with retweets and mentions relation for each edge in the network. A weight value represents the degree to which one user can cooperate with other users.

In these experiments, the weights are calculated using Equation (3.1) utilizes the structural and content attributes by implementing Algorithm (3.7). A new weights are determined by utilizing the weight of retweets and mentions with topics and user's sentiment. Figure (4.8) has shown a new weighted network graph sample, where the new weights are calculated using structural and content attributes. The figure has stated the weights associated with the user's topics, sentiment, retweet, and mention. This new weight network will be used in discover user's communities.

The community detection is started after obtaining the new weight network by applying the Algorithm (3.8). The Leiden algorithm starts by creating a singleton node partition. Then, nodes are moved locally from one community to another to find a partition. The refinement is being fine-tuned partition. Finally, network aggregation is based on this refined partition. All procedures are repeated until no further improvements are possible.

The experiments measured the improved Leiden algorithm by determining modularity and a Constant Potts Model (CPM). Modularity evaluates how well a network can be divided into communities that have more connections among nodes inside groups but fewer connections among nodes in different groups.

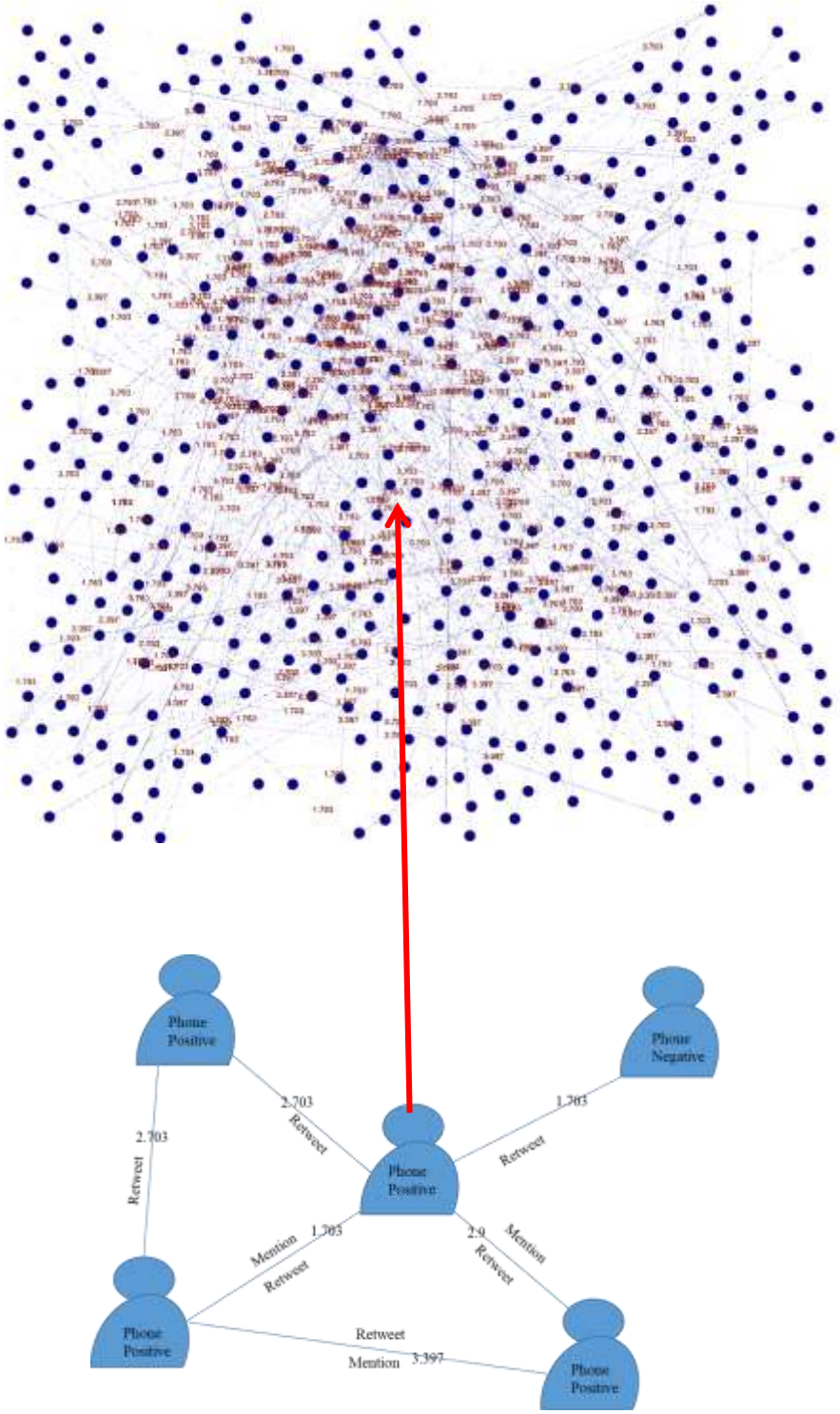


Figure 4. 8 Enhanced Detecting communities

The improve Leiden community detection was applied on tweets network. The performance of improved Leiden community detection was tested on networks by comparing the algorithm's results on five different types of networks that contain user content and attributes. The parameters were adjusted for modularity with resolution parameter ($\gamma = 1$) and CPM constant parameter ($\gamma =$ 0.01, 0.001). Table (4.14) are shown the result of these experiments that have tested the proposed model for structural and content networks. The result shows improvement in quality of community detection for both modularity and CPM metrics.

Table 4. 14. Leiden Community Detection Result of Dataset 1

Type	Modularity	CPM γ=0.01	CPM γ=0.001
Retweet Network	0.7984	0.7514	0.9133
Retweet and Mention Network	0.8302	0.8123	0.9495
Retweet and Topic Network	0.7851	0.8106	0.9538
Retweet, Mention and Topic Network	0.8136	0.8361	0.9574
Retweet, Mention and Topic, Sentiment Network	**0.8234**	**0.8488**	**0.9637**

The network that incorporates retweet weights and mentions with topics and sentiment has the higher values for modularity (0.8234) and CPM (0.8488 and 0.9637) metrics. In contrast, the structural network that depends on link relation shows less quality value for the improved Leiden method. The retweet network shows less quality value for both modularity

and CPM metrics. The retweet and mention network combination show better results than the retweet network because of the increased weight of edges with mention edges. The network that incorporates retweet, mention, and the topic shows better results than the retweet network and retweet and mention network because of the increased weight of edges with mention edges and topics. The network that incorporates retweet and mention with the topic and sentiment shows better results than other network because of the increased weight of edges with mention edges, topics, and sentiments. Thus, content and Twitter features help enhance the Leiden method results.

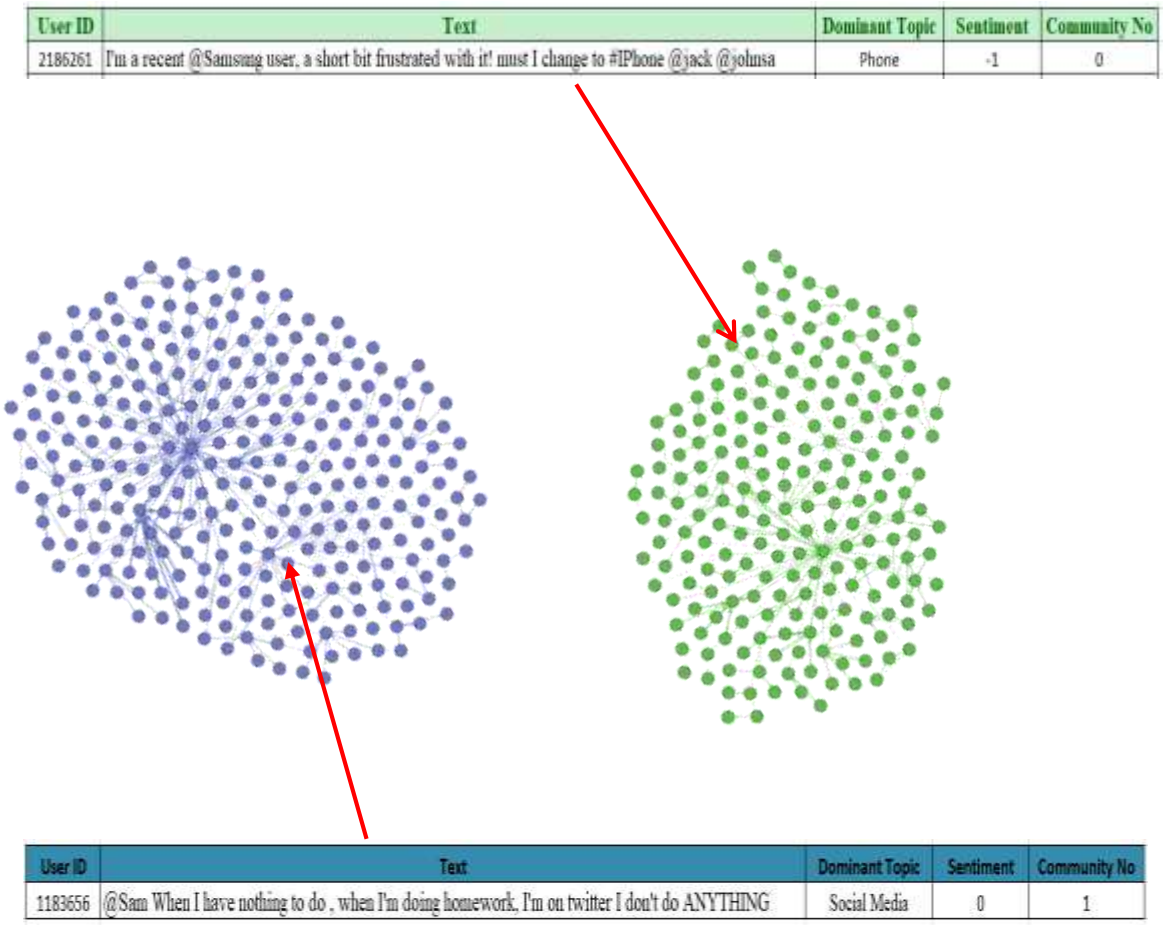


Figure 4. 9 Community Detection Result Details

The modularity showed less change in their value than CPM because of resolution limits which may cluster smaller communities into larger communities, while CPM has not contained this problem. Thus, high CPM values are easier to understand. The retweet network has shown less value for modularity and CPM because it only depends on link relationships, which ignores other attributes that may help improve quality. The retweet and topics network performs close results to the retweet, mention, and topic network because a small number of mentions exist between users in dataset. Update edge weight values with additional attributes allowed community detection to discover more accurate communities representing user's interactions and discussing topics between users and their sentiments. Every node in the community has not only the community number of nodes, but it also has helpful information about user topics and opinions. Figure (4.9) shows the result of the two communities. The user in community 0 writes about the phone and has a negative opinion about phone. While the user in community 1 writes about social media and has a neutral opinion about social media. This information helps researchers and companies analyze and detect users' behavior in communities.

Further, this combination will help know community sentiments about a particular topic in networks by providing users' opinions. Figure (4.10) displays the distribution of opinion in the three most significant communities. The positive users opinion is high in all three communities, while negative users opinion is low in all three communities. Also, The neutral and negative opinions are almost equal in community 1.

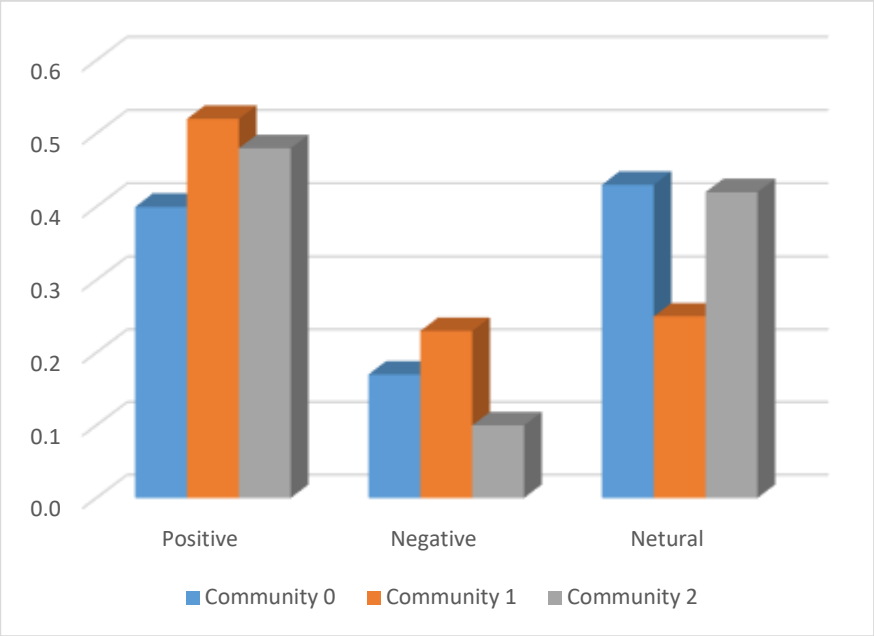


Figure 4. 10 Distribution of Sentiment in Communities

4.5 Other Experiments

The proposed model was applied on the collected dataset 2 of the Twitter network. The preprocessing step in Algorithm (3.2) was applied on tweet dataset to change the tweets to the readable format required by the next phase of the proposed model. Then, retweet and mention networks are built-in Twitter as an interactive graph representing a group of nodes connected by edges with weights assigned. These mentions and retweets are extracted in preprocessing used to build networks in Algorithm (3.3). Next, the hashtags enrichment process is applied to cleaned tweets by appending hashtags words extracted from the dataset. The enriched tweets were mapped into a numerical representation of tweets dataset called TF-IDF.

The LSA topic model was used in this experiments. Several experiments have been executed on various setups to extract latent topics by applying the LSA method in Algorithm (3.4). The experiments used k's topics between 2 and 20 to find the best cluster topics. The coherence score is used to evaluate each experiment of topic modelling. Table (4.15) has shown the result of LSA model experiments. LSA with hashtags enrichment tweets gets higher coherence value than without hashtags enrichment because adding hashtags into tweets increases word co-occurrences with rich words.

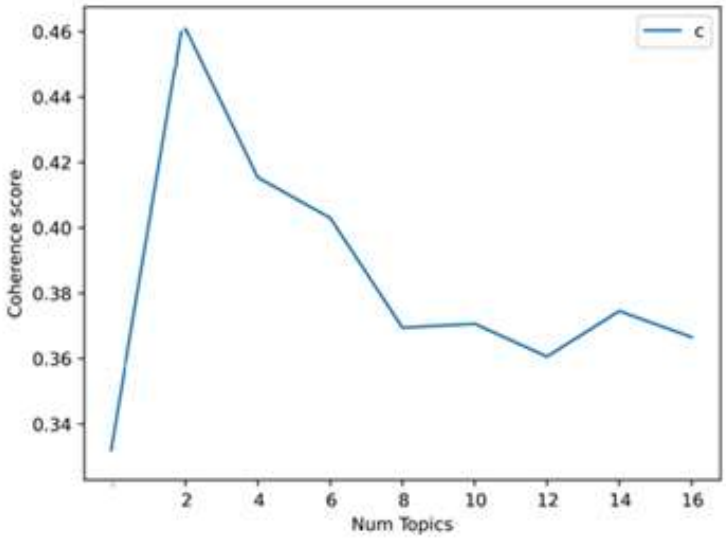


Figure 4. 11 Coherence Score of LSA for Different K

The experiments showed the best result for LSA with hashtags enrichment tweets when k = 4 with a coherence score of 0.4744 see Figure (4.11), while The best number of topics for LSA without hashtags is k = 2 with a coherence score of 0.4269. The result shows that LSA incorporates hashtags that perform better than LSA without hashtags on

most k topic numbers. Using hashtags attribute to enriched tweets helps improve the coherence score value in our LSA result from 0.4269 to 0.4744.

Table 4. 15. Coherence Score for LSA on Dataset 2

Topic No.	LSA with Hashtags	LSA without Hashtags
2	0.3229	**0.4269**
4	**0.4744**	0.385
6	0.4449	0.3679
8	0.3329	0.278
10	0.3955	0.3281
12	0.3229	0.3124
14	0.2615	0.2936
16	0.3384	0.2764
18	0.3136	0.3041

Table (4.16) has shown the topics and their words discovered by the LSA method on enriched tweets. The words in the topics were ordered according to their probability. Figure (4.12) is a word cloud that plots words when each word is scaled by its probability for an additional indication. The experiments have been executed on LSA tweets output to discover user opinion by applying the SVM method in Algorithm (3.6). The result of SVM in Figure (4.13) has shown very few users have negative opinions, while most users have a positive and neutral opinion about four topics.

Table 4. 16. LSA topics result on Enriched Tweets

Topic #0	Topic #1	Topic #3	Topic #4
anxiety	driver	school	movie
attack	car	college	vacation
time	race	course	artist
depression	ride	student	life
today	man	knowledge	cinema
people	front	teacher	picnic
thing	accident	tuition	lounge
day	insurance	university	break
stress	park	level	interest
work	finance	institute	holiday

Figure 4. 12 LSA Word Cloud for Topics on Dataset 2

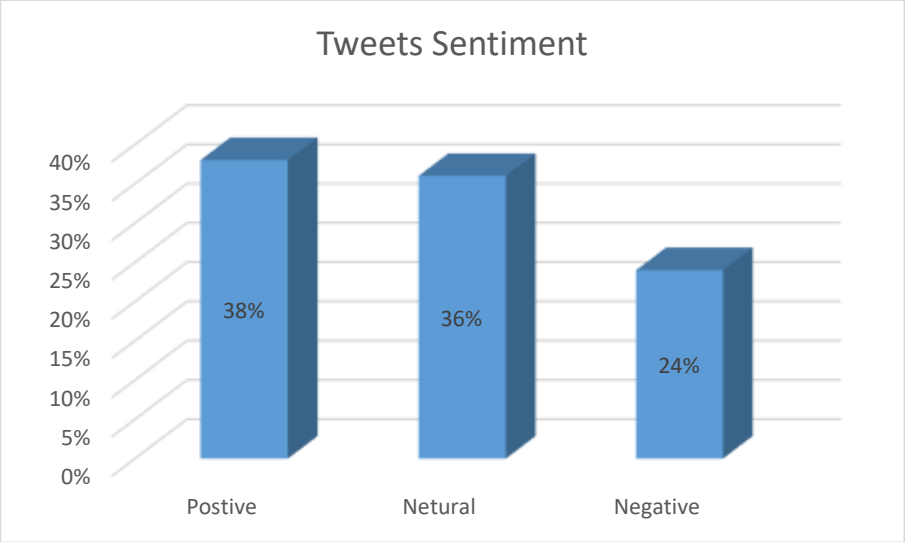


Figure 4. 13 SVM Sentiment Result on Dataset 2

A new edges network was calculated by applying the Algorithm (3.7). The Leiden community detection is started after obtaining the new weight network by applying the Algorithm (3.8). The Leiden community detection algorithm was applied on new weighted network. The performance of improved Leiden community detection was tested on Twitter networks by comparing the algorithm's results on five different types of networks that contain user content and attributes. The parameters were adjusted for modularity with resolution parameter ($\gamma = 1$) and CPM constant parameter ($\gamma =$ 0.01, 0.001). Table (4.17) is shown the result of experiments that have tested the improved Leiden method for structural and content networks. The result shows improvement in quality of community detection for both modularity and CPM. The network that incorporates retweet weights and mentions with topics and sentiment has the higher values for modularity (0.9828) and CPM (0.7395 and 0.9537)

metrics. In contrast, the structural network that depends on link relation shows less quality value for the improved Leiden method.

Table 4. 17. Community Detection Result on Dataset2

Type	Modularity	CPM γ=0.01	CPM γ=0.001
Retweet Network	0.9824	0.7174	0.9410
Retweet and Mention Network	0.9825	0.7337	0.9525
Retweet and Topic Network	0.9826	0.7359	0.9530
Retweet, Mention and Topic Network	0.9826	0.7364	0.9530
Retweet, Mention and Topic, Sentiment Network	**0.9828**	**0.7395**	**0.9537**

4.6 Benchmark Dataset

For more general, The proposed model was tested on a benchmark dataset. Zachary karate club network is a standard network dataset. It contains 34 nodes and 78 edges assigned a weight one. This dataset only represents a structural network that does not accumulate text information. The performance of proposal was tested on Zachary Karate Club benchmark network by comparing the results of the improved Leiden community detection on two types of networks. We suppose that the Zachary network includes data about users. The weights between edges have been updated randomly for 30 link relations in experiments. The parameters were adjusted for modularity with resolution parameter ($\gamma = 1$) and CPM constant parameter ($\gamma = 0.05$).

Table 4. 18. Zachary Network Result

Type	Modularity	CPM γ=0.05	CPM γ=0.1
Zachary Network	0.4198	0.6865	0.5282
New Zachary Network	**0.4647**	**0.8295**	**0.7447**

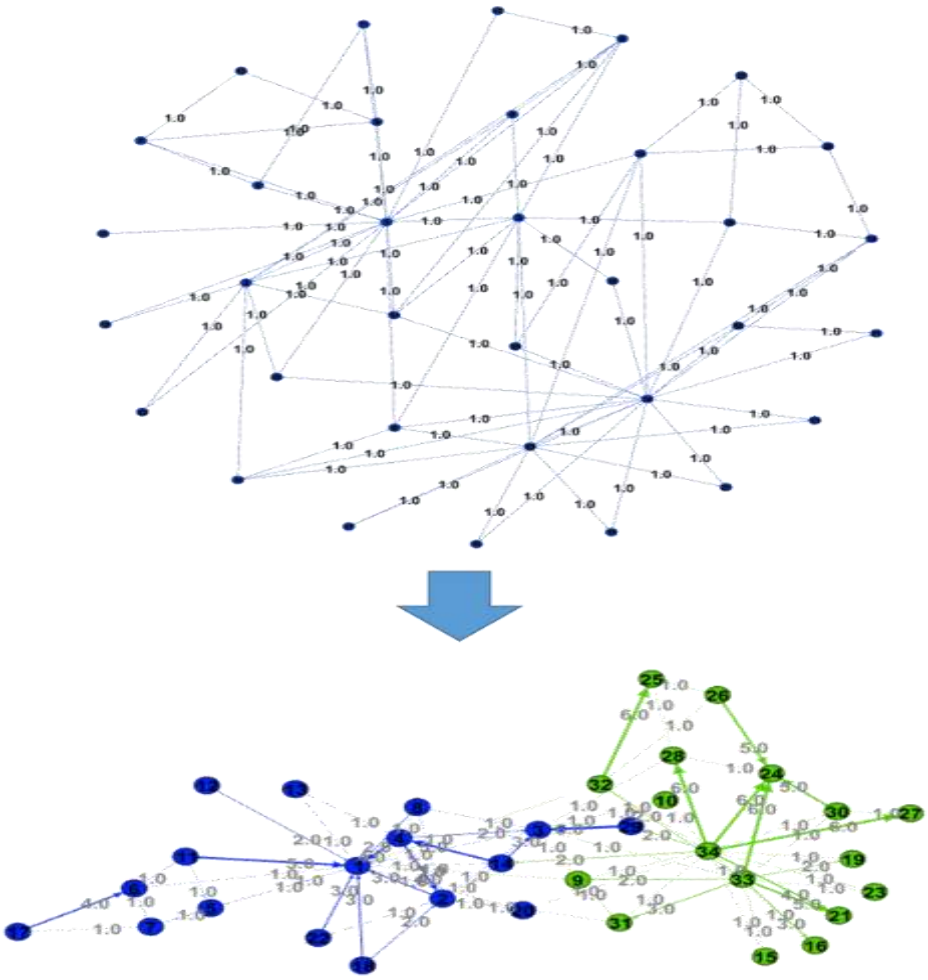


Figure 4. 14 Zachary Communities

Table (4.18) is shown the result of experiments that have tested the improved Leiden method for structural Zachary network and update Zachary network. The result shows improvement in quality of enhanced Leiden community detection for modularity and CPM when updating edges weight. A new weights network increases the quality of enhanced Leiden algorithm for modularity from 0.4198 in link relationship to 0.4647 and CPM from 0.5282 and 0.6865 in link relationship to 0.7447 and 0.8295. Figure (4.14) was shown the Zachary network has updated the weights of the edges. Then, communities were detected using the enhanced Leiden method.

4.7 Results Comparison

The proposed enhancement work has been compared on the dataset of tweets for topic modelling and community detection.

4.7.1 LSA and LDA Topic Modelling

LSA is compared with the LDA topic model algorithm using tweets to find latent topics. Each method executes for several numbers of k depending on the dataset. The topics were used between k = 2 and 20, and the coherence score was calculated for each technique with k topics. LSA and LDA clustering methods are applied to the TF-IDF built from Twitter dataset. The results of LSA and LDA methods with hashtags and without hashtags are listed in Table (4.19) and Figure (4.15). The best number of topics for LSA with hashtags is a k = 4 with a coherence score of 0.4744, while the best number of topics for LSA without hashtags is k = 2 with a coherence score of 0.4269. For the LDA method, the best number of topics

with hashtags is 16 topics with a coherence score of 0.6036, while the best number of topics without hashtags is 14 topics with a coherence score of 0.6047. The LDA coherence score value has increased with the number of topics. Using hashtags attribute to enriched tweets helps improve the coherence score value in LSA result, while LDA performs better without incorporating hashtags. LDA has a higher coherence score value than LSA due to the Dirichlet process finding more topics in the text. The words are still duplicated in many topics, and some topics can cluster into the same topic rather than be put into multiple topics that increase coherence value. In contrast, LSA is a semantic structure method that collects words to share the same subject into the same topic. Therefore, the number of k topics in LSA is less than LDA due to collecting all tweets sharing the same topics into one general topic. Thus, LSA coherence score values are less than LDA in the result.

Table 4. 19. Comparison between LSA and LDA Results

Topic No.	LSA with Hashtags	LSA without Hashtags	LDA with Hashtags	LDA without Hashtags
2	0.3229	**0.4269**	0.285	0.357
4	**0.4744**	0.385	0.3788	0.3279
6	0.4449	0.3679	0.4304	0.4368
8	0.3329	0.278	0.4856	0.5077
10	0.3955	0.3281	0.5047	0.5716
12	0.3229	0.3124	0.5944	0.5574
14	0.2615	0.2936	0.5867	**0.6047**
16	0.3384	0.2764	**0.6036**	0.6023
18	0.3136	0.3041	0.6027	0.6004

Figure 4. 15 LSA and LDA coherence Score Results

4.7.2 Leiden and Louvain Community Detection Algorithms

The proposed model and Louvain were compared with different networks containing user content and attributes. The results of the experiments are shown in Table (4.20). The improve Leiden community detection acts better than the Louvain community detection regarding the clustering quality obtained. It determinates substantially better community than Louvain in all networks. The proposed model outperforms Louvain for both modularity and CPM in all experiments.

The results have shown the network combines retweet, mention, topic, and sentiment get high-quality function for improved Leiden modularity 0.82 and CPM 0.96. The differences between the results of the two methods aren't significant, probably because both ways try to find partitions with optimal clusters. Therefore, the quality function is near-optimal quality, making results close to each other. The new weight calculation is essential to improve the quality of communities detection methods. The Leiden and Louvain method improved when combined content and structure data to calculate new weights.

Table 4. 20. Comparison Result of Community Detection

	resolution parameter γ= 0.001			
Type	**Leiden Modularity**	**Leiden CPM**	**Louvain Modularity**	**Louvain CPM**
Retweet Network	0.7984	0.9133	0.7864	0.9096
Retweet and Mention Network	0.8302	0.9495	0.8297	0.9342
Retweet and Topic Network	0.7851	0.9538	0.7742	0.9425
Retweet, Mention and Topic Network	0.8136	0.9574	0.8026	0.9401
Retweet, Mention and Topic, Sentiment Network	**0.8234**	**0.9637**	**0.8124**	**0.9544**

Chapter Five

Conclusion and Future Works

5.1 Conclusion

Nowadays, the social network has overgrown because people share their opinions, feelings, ideas, and life. They tend to create communities to share ideas, topics, or events. Communities detection is an attractive area for several researchers in social networks. Most current research only concentrate on link relation between users that disregards other attributes to enhance the quality of community detection. Therefore, It is important to improve and develop the community detection algorithms by adding more useful information on communities rather than only depending on network connections. Several conclusions have been found through the development and implementation of the proposed model.

1- There is no existing benchmark dataset available in public that has user content and relationships in the same dataset. Therefore in this work construct a dataset with text content and link relation by downloading from Twitter to build valuable data.

2- Social network data has many meaningless and irrelevant terms that may cause unreasonable results. In this work, many preprocessing techniques were implemented to clean the Twitter social media datasets because of the nature of Twitter posts.

3- Hashtags attributes in Twitter social network have store important content that help to enrich and discover latent topics in users content. Hashtags improve the performance of the LSA topic method.

4- Other social network attributes (mentions, retweets, and replays) are used to build interaction networks utilized to detect communities.

5- Topics and sentiment results with mentions and retweets relation are used to calculate a new weight between nodes to build a new network.

6- The present model has improved Leiden community detection quality for both modularity and CPM when using new weights networks that incorporate content and attributes of the Twitter social network.

7- Thus, incorporating helps enhance the community detection and add more information that allows companies and businesses to get details about their product and know which topics are discussed between users and their sentiments about a particular topic in networks by providing information about nodes and communities.

5.2 Future Works

For future work, some suggestions can be viewed as follows:

1- The proposed model can apply to other social network datasets like Facebook or Instagram that have different characteristics.

2- Testing proposed model with another language like Arabic after editing work with appropriate change.

3- Exploit social network emoticons and abbreviations rather than selecting only complete terms from the posts, which sometimes

donate to the texts emotion and sentiment to get high classification performance.

4- The proposed model generalizes with a dynamic community that can adjust the influence changes in the graph representation based on users behavior.

5- Communities results can utilize to understand and finding influencer users, whether they are users with a high connection with other users or those with who they agree about a specific topic or opinion to see what indicates someone to be significant in the community.

References

REFERENCES

[1] A. Culotta, "Towards detecting influenza epidemics by analyzing Twitter messages," in *Proceedings of the first workshop on social media analytics*, 2010, pp. 115–122.

[2] "• Twitter by the Numbers (2021): Stats, Demographics & Fun Facts." https://www.omnicoreagency.com/twitter-statistics/ (accessed Nov. 13, 2021).

[3] E. Jokar and M. Mosleh, "Community detection in social networks based on improved Label Propagation Algorithm and balanced link density," *Phys. Lett. A*, vol. 383, no. 8, pp. 718–727, 2019.

[4] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Min. Knowl. Discov.*, vol. 24, no. 3, pp. 515–554, 2012.

[5] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, 2002.

[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. theory Exp.*, vol. 2008, no. 10, p. P10008, 2008.

[7] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E*, vol. 76, no. 3, p. 36106, 2007.

[8] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Natl. Acad. Sci.*, vol. 105, no. 4, pp. 1118–1123, 2008.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[11] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 50–57.

[12] A. Anantharaman, A. Jadiya, C. T. S. Siri, B. N. V. S. Adikar, and B. Mohan, “Performance evaluation of topic modeling algorithms for text classification,” in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 704–708.

[13] S. Ji, C. P. Yu, S. Fung, S. Pan, and G. Long, “Supervised learning for suicidal ideation detection in online user content,” *Complexity*, vol. 2018, 2018.

[14] N. S. Sattar and S. Arifuzzaman, “COVID-19 vaccination awareness and aftermath: Public sentiment analysis on Twitter data and vaccinated population prediction in the USA,” *Appl. Sci.*, vol. 11, no. 13, p. 6128, 2021.

[15] Z. Qu, J. Yang, X. Wang, and S. Yin, “Combining Link and Content for Community Detection in Social Networks,” in *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2018, pp. 607–610.

[16] A. Kanavos, I. Perikos, I. Hatzilygeroudis, and A. Tsakalidis, “Emotional community detection in social networks,” *Comput. Electr. Eng.*, vol. 65, pp. 449–460, 2018.

[17] V. A. Traag, L. Waltman, and N. J. Van Eck, “From Louvain to Leiden: guaranteeing well-connected communities,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, 2019.

[18] A. Clauset, M. E. J. Newman, and C. Moore, “Finding community structure in very large networks,” *Phys. Rev. E*, vol. 70, no. 6, p. 66111, 2004.

[19] X. You, Y. Ma, and Z. Liu, “A three-stage algorithm on community detection in social networks,” *Knowledge-Based Syst.*, vol. 187, p. 104822, 2020.

[20] G. Bello-Orgaz, J. Hernandez-Castro, and D. Camacho, “Detecting discussion communities on vaccination in twitter,” *Futur. Gener. Comput. Syst.*, vol. 66, pp. 125–136, 2017.

[21] H. Zhou, J. Li, J. Li, F. Zhang, and Y. Cui, “A graph clustering method for community detection in complex networks,” *Phys. A Stat. Mech. Its Appl.*, vol. 469, pp. 551–562, 2017.

[22] M. Coscia, “Discovering communities of community discovery,” in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2019, pp. 1–8.

[23] C. D. G. Linhares, J. R. Ponciano, F. S. F. Pereira, L. E. C. Rocha, J. G. S. Paiva, and B. A. N. Travençolo, “Visual analysis for evaluation of community detection algorithms.,” *Multimed. Tools Appl.*, vol. 79, 2020.

[24] T. Ma, Q. Liu, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, “LGIEM: Global and local node influence based community detection,” *Futur. Gener. Comput. Syst.*, vol. 105, pp. 533–546, 2020.

[25] X. Bai, P. Yang, and X. Shi, “An overlapping community detection algorithm based on density peaks,” *Neurocomputing*, vol. 226, pp. 7–15, 2017.

[26] D. L. Sánchez, J. Revuelta, F. De la Prieta, A. B. Gil-González, and C. Dang, “Twitter user clustering based on their preferences and the Louvain algorithm,” in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, 2016, pp. 349–356.

[27] C. Li, J. Bai, Z. Wenjun, and Y. Xihao, “Community detection using hierarchical clustering based on edge-weighted similarity in cloud environment,” *Inf. Process. Manag.*, vol. 56, no. 1, pp. 91–109, 2019.

[28] N. Alduaiji and A. Datta, “An empirical study on sentiments in twitter communities,” in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2018, pp. 1166–1172.

[29] C. He, Y. Tang, H. Liu, X. Fei, H. Li, and S. Liu, “A robust multi-view clustering method for community detection combining link and content information,” *Phys. A Stat. Mech. its Appl.*, vol. 514, pp. 396–411, 2019.

[30] M. Bakillah, R.-Y. Li, and S. H. L. Liang, “Geo-located community detection in Twitter with enhanced fast-greedy optimization of modularity: the case study of typhoon Haiyan,” *Int. J. Geogr. Inf. Sci.*, vol. 29, no. 2, pp.

258–279, 2015.

[31] B. R. C. Amor, S. I. Vuik, R. Callahan, A. Darzi, S. N. Yaliraki, and M. Barahona, “Community detection and role identification in directed networks: understanding the Twitter network of the care. data debate,” in *Dynamic networks and cyber-security*, World Scientific, 2016, pp. 111–136.

[32] S. Bhatt *et al.*, “Knowledge graph enhanced community detection and characterization,” in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 51–59.

[33] M. Benabdelkrim, J. Savinien, and C. Robardet, “Finding interest groups from Twitter lists,” in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 1885–1887.

[34] T. B. N. Hoang, “Topical Community Detection: an Embedding User and Content Similarity Method,” 2020.

[35] D. Surian, D. Q. Nguyen, G. Kennedy, M. Johnson, E. Coiera, and A. G. Dunn, “Characterizing Twitter discussions about HPV vaccines using topic modeling and community detection,” *J. Med. Internet Res.*, vol. 18, no. 8, p. e232, 2016.

[36] D. Camacho, Á. Panizo-LLedot, G. Bello-Orgaz, A. Gonzalez-Pardo, and E. Cambria, “The four dimensions of social network analysis: An overview of research methods, applications, and software tools,” *Inf. Fusion*, vol. 63, pp. 88–120, 2020.

[37] P. Bedi and C. Sharma, “Community detection in social networks,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 6, no. 3, pp. 115–135, 2016.

[38] D. T. Schroeder, K. Pogorelov, and J. Langguth, “Fact: a framework for analysis and capture of twitter graphs,” in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 2019, pp. 134–141.

[39] N. Agarwal, N. Dokoohaki, and S. Tokdemir, *Emerging research challenges and opportunities in computational social network analysis and mining*. Springer, 2019.

[40] 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.

[41] "About different types of Tweets." https://help.twitter.com/en/using-twitter/types-of-tweets (accessed Nov. 14, 2021).

[42] P. K. Novak, L. De Amicis, and I. Mozetič, "Impact investing market on Twitter: influential users and communities," *Appl. Netw. Sci.*, vol. 3, no. 1, p. 40, 2018, doi: 10.1007/s41109-018-0097-9.

[43] K. Sailunaz and R. Alhajj, "Emotion and sentiment analysis from Twitter text," *J. Comput. Sci.*, vol. 36, p. 101003, 2019.

[44] X. Chen, D. Zou, G. Cheng, and H. Xie, "Detecting latent topics and trends in educational technologies over four decades using structural topic modeling: A retrospective of all volumes of Computers & Education," *Comput. Educ.*, vol. 151, p. 103855, 2020.

[45] E. Cambria, H. Wang, and B. White, "Guest editorial: Big social data analysis," *Knowledge-based Syst.*, vol. 69, no. 1, pp. 1–2, 2014.

[46] M. Asif, A. Ishtiaq, H. Ahmad, H. Aljuaid, and J. Shah, "Sentiment analysis of extremism in social media from textual information," *Telemat. Informatics*, vol. 48, p. 101345, 2020.

[47] I. Habernal, T. Ptáček, and J. Steinberger, "Supervised sentiment analysis in Czech social media," *Inf. Process. Manag.*, vol. 50, no. 5, pp. 693–707, 2014.

[48] M. Ghiassi and S. Lee, "A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach," *Expert Syst. Appl.*, vol. 106, pp. 197–216, 2018.

[49] B. Dahal, S. A. P. Kumar, and Z. Li, "Topic modeling and sentiment analysis of global climate change tweets," *Soc. Netw. Anal. Min.*, vol. 9, no. 1, pp. 1–20, 2019.

[50] A. Steinskog, J. Therkelsen, and B. Gambäck, "Twitter topic modeling by tweet aggregation," in *Proceedings of the 21st nordic conference on computational linguistics*, 2017, pp. 77–86.

[51] R. Nugroho, C. Paris, S. Nepal, J. Yang, and W. Zhao, “A survey of recent methods on deriving topics from Twitter: algorithm to evaluation,” *Knowl. Inf. Syst.*, pp. 1–35, 2020.

[52] S. A. Curiskis, B. Drake, T. R. Osborn, and P. J. Kennedy, “An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit,” *Inf. Process. Manag.*, vol. 57, no. 2, p. 102034, 2020.

[53] R. Albalawi, T. H. Yeap, and M. Benyoucef, “Using topic modeling methods for short-text data: A comparative analysis,” *Front. Artif. Intell.*, vol. 3, p. 42, 2020.

[54] M. Bilgin and İ. F. Şentürk, “Sentiment analysis on Twitter data with semi-supervised Doc2Vec,” in *2017 international conference on computer science and engineering (UBMK)*, 2017, pp. 661–666.

[55] G. Giasemidis, N. Kaplis, I. Agrafiotis, and J. R. C. Nurse, “A semi-supervised approach to message stance classification,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 1, pp. 1–11, 2018.

[56] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–130, 2009.

[57] S. Qomariyah, N. Iriawan, and K. Fithriasari, “Topic modeling twitter data using latent dirichlet allocation and latent semantic analysis,” in *AIP conference proceedings*, 2019, vol. 2194, no. 1, p. 20093.

[58] T. Vo *et al.*, “Crime rate detection using social media of different crime locations and Twitter part-of-speech tagger with Brown clustering,” *J. Intell. Fuzzy Syst.*, vol. 38, pp. 4287–4299, 2020, doi: 10.3233/JIFS-190870.

[59] H. Christian, M. P. Agus, and D. Suhartono, “Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF),” *ComTech Comput. Math. Eng. Appl.*, vol. 7, no. 4, pp. 285–294, 2016.

[60] A. A. Hakim, A. Erwin, K. I. Eng, M. Galinium, and W. Muliady, “Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach,” in *2014 6th international conference on information technology and*

electrical engineering (ICITEE), 2014, pp. 1–4.

[61] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 273–292, 2019.

[62] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Process.*, vol. 25, no. 2–3, pp. 259–284, 1998.

[63] S. Santilli, L. Nota, and G. Pilato, "A Comparison on the Use of LSA and LDA in Psychology Analysis on 'Courage' Definitions," *Int. J. Semant. Comput.*, vol. 11, no. 03, pp. 373–389, 2017.

[64] B. Pratama *et al.*, "Sentiment Analysis of the Indonesian Police Mobile Brigade Corps Based on Twitter Posts Using the SVM and NB Methods," in *Journal of Physics: Conference Series*, 2019, vol. 1201, no. 1, p. 12038.

[65] B. Gaye, D. Zhang, and A. Wulamu, "Improvement of Support Vector Machine Algorithm in Big Data Background," *Math. Probl. Eng.*, vol. 2021, 2021.

[66] S. Tabassum, F. S. F. Pereira, S. Fernandes, and J. Gama, "Social network analysis: An overview," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 5, p. e1256, 2018.

[67] N. Alduaiji, A. Datta, and J. Li, "Influence propagation model for clique-based community detection in social networks," *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 2, pp. 563–575, 2018.

[68] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 26113, 2004.

[69] M. Azaouzi, D. Rhouma, and L. Ben Romdhane, "Community detection in large-scale social networks: state-of-the-art and future directions," *Soc. Netw. Anal. Min.*, vol. 9, no. 1, pp. 1–32, 2019.

[70] Z. Liu and Y. Ma, "A divide and agglomerate algorithm for community detection in social networks," *Inf. Sci. (Ny).*, vol. 482, pp. 321–333, 2019.

[71] S. Rahimi, A. Abdollahpouri, and P. Moradi, "A multi-objective particle swarm optimization algorithm for community detection in complex networks," *Swarm Evol. Comput.*, vol. 39, pp. 297–309, 2018.

[72] A. Mockus, D. Spinellis, Z. Kotti, and G. J. Dusing, “A complete set of related git repositories identified via community detection approaches based on shared commits,” in *Proceedings of the 17th International Conference on Mining Software Repositories*, 2020, pp. 513–517.

[73] Y. Chen and J. W. Baker, “Community detection in spatial correlation graphs: Application to non-stationary ground motion modeling,” *Comput. Geosci.*, vol. 154, p. 104779, 2021.

[74] X. Que, F. Checconi, F. Petrini, and J. A. Gunnels, “Scalable community detection with the louvain algorithm,” in *2015 IEEE International Parallel and Distributed Processing Symposium*, 2015, pp. 28–37.

[75] X. Yuan, R. J. Schuchard, and A. T. Crooks, “Examining emergent communities and social bots within the polarized online vaccination debate in Twitter,” *Soc. media+ Soc.*, vol. 5, no. 3, p. 2056305119865465, 2019.

[76] T. M. Consortium, “A single cell transcriptomic atlas characterizes aging tissues in the mouse,” *Nature*, vol. 583, no. 7817, p. 590, 2020.

[77] Q. Zhang *et al.*, “Landscape and dynamics of single immune cells in hepatocellular carcinoma,” *Cell*, vol. 179, no. 4, pp. 829–845, 2019.

[78] A. Panizo-LLedot, J. Torregrosa, G. Bello-Orgaz, J. Thorburn, and D. Camacho, “Describing alt-right communities and their discourse on twitter during the 2018 us mid-term elections,” in *International conference on complex networks and their applications*, 2019, pp. 427–439.

[79] Z. Tong and H. Zhang, “A text mining research based on LDA topic modelling,” in *International Conference on Computer Science, Engineering and Information Technology*, 2016, pp. 201–210.

[80] Z. Jianqiang and G. Xiaolin, “Comparison research on text pre-processing methods on twitter sentiment analysis,” *IEEE Access*, vol. 5, pp. 2870–2879, 2017.

[81] W. Cui *et al.*, “Extended search method based on a semantic hashtag graph combining social and conceptual information,” *World Wide Web*, vol. 22, no. 6, pp. 2589–2610, 2019.

[82] M. Prateek and V. Vasudeva, “Improved topic models for social media via

community detection using user interaction and content similarity," in *2016 international FRUCT conference on intelligence, social media and web (ISMW FRUCT)*, 2016, pp. 1–7.

[83] P. Chunaev, I. Nuzhdenko, and K. Bochenina, "Community detection in attributed social networks: a unified weight-based model and its regimes," in *2019 International Conference on Data Mining Workshops (ICDMW)*, 2019, pp. 455–464.

[84] S. J. Blair, Y. Bi, and M. D. Mulvenna, "Aggregated topic models for increasing social media topic coherence," *Appl. Intell.*, vol. 50, no. 1, pp. 138–156, 2020.

[85] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler, "Exploring topic coherence over many models and many topics," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 952–961.

[86] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The impact of features extraction on the sentiment analysis," *Procedia Comput. Sci.*, vol. 152, pp. 341–348, 2019.

0.9637. في حين أن حساب شبكة الأوزان الجديدة يزيد من جودة طريقة Leiden على مجموعة بيانات نادي زكاري للكاراتيه للوحدات النمطية من 0.4198 في علاقة ارتباط بـ 0.4647 و CPM من 0.5282 و 0.6865 في علاقة ارتباط بـ 0.7447 و 0.8295.

الخلاصة

في الوقت الحاضر ، توسعت وسائل التواصل الاجتماعي (أو الشبكات الاجتماعية) بسرعة كبيرة. يستخدم الناس وسائل التواصل الاجتماعي لمشاركة آرائهم وأفكارهم ومشاعرهم. ينشرون أنواعًا مختلفة من المحتوى مثل النصوص والصور ومقاطع الفيديو ويتفاعلون فيما بينهم باستخدام مميزات الشبكة الاجتماعية مثل الإشارات وعلامات التصنيف (هاشتاك) وإعادة النشر والإعجابات. يميل الناس إلى بناء مجموعات استناداً على العلاقات مع الآخرين ، مثل التفاعل أو نشر نفس المواضيع أو الآراء. تم تقديم اكتشاف المجتمعات كطريقة للعثور على مجتمعات في الشبكات الاجتماعية عن طريق تقسيم الشبكة إلى مجموعات من المستخدمين المتصلين بشكل وثيق. يعد اكتشاف المجتمع تحديًا كبيرًا عند تحليل وسائل التواصل الاجتماعي بسبب هيكلها وتنوع مواردها. تركز الدراسات الحالية على شبكة هيكلية تعتمد فقط على الروابط بين المستخدمين لاكتشاف المجتمعات. قد لا تحلل هذه الأعمال المجتمعات جيدًا لأنها تعتمد فقط على الروابط والعقد في الشبكة.

تم تطوير النموذج المقترح من خلال استغلال تقنيات مختلفة لتعزيز طريقة اكتشاف مجتمع Leiden. تستغل التقنية الأولى خوارزمية التحليل الدلالي الكامن (LSA) مع علامات التصنيف (هاشتاك) لاكتشاف مواضيع المستخدمين لتحسين أداء اكتشاف المجتمع. يؤدي استخدام LSA مع اكتشاف المجتمع إلى منح المجتمعات مزيدًا من المعلومات السياقية. ثانيًا ، استخدام سمات الإشارة وإعادة التغريد لبناء شبكات تفاعل. ثالثًا ، يتم استخدام نظام الدعم الالي (SVM) للحصول على آراء المستخدمين حول موضوع معين. رابعًا ، تم استغلال الموضوع والشعور وإعادة التغريد مع الإشارة لحساب شبكة ذات اوزان جديدة. أخيرًا ، يمكن لشبكة ذات اوزان جديدة أن تعزز جودة للوحدات النمطية ونموذج بوتس الثابت (CPM) لطرق اكتشاف المجتمع.

تم إجراء العديد من التجارب على Twitter وشبكات معيارية مختلفة. أظهرت النتائج تحسنًا في جودة اكتشاف المجتمع للنمطية والتكلفة لكل ألف ظهور وCPM عندما تستغل المواضيع ، والمشاعر ، وإعادة التغريد ، الاشارة للحصول على شبكة ذات اوزان جديدة. يعمل حساب شبكة الأوزان الجديدة على زيادة جودة طريقة Leiden للوحدات النمطية من 0.7984 في علاقة الارتباط إلى 0.8234 و CPM من 0.7514 و 0.8488 في علاقة الارتباط إلى 0.9133 و

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات - قسم البرمجيات

تحسين اكتشاف المجتمعات في وسائل التواصل الاجتماعي تويتر باستخدام نمذجة المواضيع وتحليل المشاعر

اطروحة

مقدمة إلى مجلس كلية تكنولوجيا المعلومات - جامعة بابل كجزء
من متطلبات درجة دكتوراه فلسفة في تكنولوجيا المعلومات - برمجيات

من قبل
حيدر ماجد عبد الحميد حسين

بأشراف
أ.د. غيداء عبد الحسين السلطاني

1443 هـ

2022 م