COLLABORATIVE DETECTION OF CYBERBULLYING BEHAVIOR IN TWITTER DATA

by

Amrita Mangaonkar

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Computer Sciences

Indianapolis, Indiana

August 2017

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Rajeev Raje, Chair

Department of Computer and Information Science

Dr. Mihran Tuceryan

Department of Computer and Information Science

Dr. Xia Ning

Department of Computer and Information Science

Approved by:

Dr. Shiaofen Fang

Head of the Graduate Program

Dedicated to loving memory of my grandfather.

ACKNOWLEDGMENTS

I would like to thank my professor and advisor, Dr. Rajeev R. Raje, for his support and guidance throughout the course of my graduate study and this research. I am truly grateful for his encouragement and moral support. I would also like to thank the members of my advisory committee members, Prof. Mihran Tuceryan and Prof. Xia Ning for their valuable suggestions and constructive feedback.

I would like to thank Ms. Amanda Boyle and Ms. Nicole Wittlief for providing their valuable input on my dissertation. Additionally, I am grateful to my colleagues Lahiru, Dimuthu and Allenoush for their support. Special thanks to Dimuthu for letting me use his work for validation of my work.

I am also thankful to IUPUI, and the entire staff of the Department of Computer and Information Science for providing me the opportunity to study and carry out research and for all the staff for their tremendous support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Author: Mangaonkar, Amrita, P. MS
Institution: Purdue University
Degree Received: August 2017
Title: Collaborative Detection of Cyberbullying Behavior in Twitter Data
Major Professor: Rajeev Raje

As the size of Twitter© data is increasing, so are undesirable behaviors of its users. One such undesirable behavior is cyberbullying, which could lead to catastrophic consequences. Hence, it is critical to efficiently detect cyberbullying behavior by analyzing tweets, in real-time if possible. Prevalent approaches to identifying cyberbullying are mainly stand-alone, and thus, are time-consuming. This thesis proposes a new approach called distributed-collaborative approach for cyberbullying detection. It contains a network of detection nodes, each of which is independent and capable of classifying tweets it receives. These detection nodes collaborate with each other in case they need help in classifying a given tweet. The study empirically evaluates various collaborative patterns, and it assesses the performance of each pattern in detail. Results indicate an improvement in recall and precision of the detection mechanism over the stand- alone paradigm. Further, this research analyzes scalability of the approach by increasing the number of nodes in the network. The empirical results obtained from experimentation show that the system is scalable. The study performed also incorporates the experiments that analyze behavior distributed-collaborative approach in case of failures in the system. Additionally, the proposed thesis tests this approach on a different domain, such as politics, to explore the possibility of the generalization of results.

1. INTRODUCTION

In the past decade, the world has seen many revolutions that were made possible by Social Media. It is an extremely influential innovation of our time, and is a great way to expand the boundaries of one's experiences and become socially active. However, social media is a double-sided weapon. A lot of anti-social behavior is observed on social media, including cyber-stalking, cyber-bullying, and cyber-harassment. These forms of stalking, bullying, and harassment have now become part of growing up. Moreover, this is not limited to children and young adults; anybody can be a victim.

1.1 Cyberbullying

Cyberbullying is formally defined as "willful and repeated harm inflicted through the use of computers, cell phones, and other electronic devices" (Patchin, J. W., 2014). In short, bullies typically exploit the use of electronic communication for harassing people. This harassment may be motivated by anger, frustration, revenge, or from a basic desire to control others and feel more powerful (Why do kids, n.d.). Sometimes kids cyberbully others to cope with their own low self-esteem and/or to fit in with their peers (Why Do People, n.d.). Examples of cyberbullying can include rumors sent by e-mail or posted on social media; embarrassing pictures or videos; and intimidating, insulting, and / or harassing massages posted on social networks. Once such derogatory messages, pictures, or videos are posted, it is very difficult to take these posts off the social media sites. It can happen 24 hours a day and 7 days a week, and it can even reach its victim when they are alone, outside in the school yard, or in the sports field (Patchin, J. W., 2014). Cyberbullying empowers a bully to humiliate and hurt the victim in online communities without ever getting recognized. Furthermore, the fear of getting punished or being a social pariah stops victims and bystanders from reporting incidents. This becomes a difficult problem to control.

Cyberbullying behavior is not only unacceptable, but can also lead to catastrophic consequences. Studies performed by *The Journal of Psychosocial Research on Cyberspace* show that "critical impacts occurred in almost all of the respondents' cases in the form of lower self-esteem, loneliness and disillusionment and distrust of people: The more extreme impacts were self-harm and increased aggression towards friends and family" (Šleglova, 2011). It further mentions that some of the victims developed "coping strategies." Many times, victims try to deal with cyberbullying all by themselves, which leads to a stressful situation. Additionally, it is tough for parents of the victims to know what is happening with their child online.

In order for support systems to help a victim, they need to identify the cyberbullying or signs of it at the onset. They should not expect the victim to approach them about cyberbullying. This calls for automated cyberbullying watchdog programs that could alert family members regarding cyberbullying.

1.2 Countermeasures by Social Media

Social networks provide some degree of support for the safe web experience. Tools that help to protect one's privacy are as follows:

- Twitter© provides users with the following tools ("Learn How", 2017).
    1. Allowing users to block, mute, or unfollow unwanted followers.
    2. Filters on notifications that allows users to filter out any unwanted replies or mentions from the accounts that the user do not follow.
    3. Reporting the undesirable behavior to Twitter.
    4. Warning the user about sensitive content before showing it. It works only for photos and videos.
    5. Tagging privacy for photos allows the user to decide who can and cannot tag him/her in photos.
    6. Twitter allows only 140 characters.
- Facebook ® has the following tools ("How to Report", 2017) (Basic Privacy , 2017):
    1. Facebook provides users with the assurance that, "Facebook removes bullying content when we become aware of it, and may disable the Facebook account of anyone who bullies or attacks another."
    2. Settings to block or unfriend the person bothering you.

3. Privacy settings in Facebook allow users to select the audience for their posts. It also provides users with tagging privacy. Additionally, it allows users to review post and tags before they are shared on their timeline.
4. Report the abusive content using a link included at the bottom of each post that allows the user to report abusive content.
5. Hide a story that appears in their News Feed.

However, social media sites do not provide any built in mechanisms to detect cyberbullying. They only act on incidences that are being reported. Thus, the responsibility of reporting cyberbullying still lies with victims and / or bystanders.

1.3 Need for Automatic Detection

Although tools are provided by contemporary social networking sites and laws are in place to fight cyberbullying, the majority of cyberbullying instances go unreported (Peterson, 2013). At the same time, there is no system in place for automatic detection of such behavior. Cyberbullying is one of the widely recognized problems which has a lasting impact on its victims. While healthy social behavior is the solution to this problem, social media platforms need to consider integrating tools and / or mechanisms that can help in the detection and prevention of such incidents. Therefore, to have a safer and more constructive social environment, it is necessary to design a smart network or an online patrol that will prohibit such behavior by monitoring and filtering the obscene, hateful, and improper content from social media posts.

1.4 Problem Definition and Motivation

Students in particular, and society members in general, have begun to believe that cyberbullying is “no big deal,” or that it is acceptable to harass others repeatedly if huge number of incidences go unpunished. On the other hand, if there are appropriate consequences for this action, then individuals will reconsider their actions before making such a move. Moreover, the entire school yard will understand the gravity of the situation and the fact that there are consequences to these actions. This cannot be achieved, however, by relying on the victim or bystander to report the incidents. This is the reason that most of the cyberbullying incidents are currently going unreported, and thus unpunished. Manual identification of this task would be too difficult to handle due to the sheer volume of the data being generated on social media. As suggested in the previous section, automated, round the clock, and accurate detection mechanisms are imperative for dealing with the problem of cyberbullying.

There are more than 100 social media websites; however, Facebook, Ask.FM, and Twitter are found to be the most likely sources of cyberbullying (Cyber Bullying , 2015). Therefore, this study focuses on detecting cyberbullying behavior in a publically available Twitter © dataset. Twitter is a general micro-blogging site. Registered users can read or post messages that are limited to 140 characters, referred to as "Tweets." These tweets are public by default. Twitter also enables registered users to share photos and videos. Unregistered users can read publicly available tweets. Although cyberbullying may appear in many forms, such as posting embarrassing messages, pictures, and videos, for the purposes of this thesis, detection of cyberbullying is limited to detection of textual cyberbullying in a Twitter dataset.

Various methods have been proposed for the detection of cyberbullying in a given textual content. These include Bag-of-Words (BoW), Lexical Syntactic Feature (LSF) (Chen, Y., Zhou, Y., Zhu, S., and Xu, H., 2012), and different Machine Learning-based approaches (Dinakar, K., Jones, B., Havasi, C., Lieberman, H., & Picard, R, 2012). Lexicon-based methods, such as the BoW or LSF, mainly rely on the presence of obscenities and profanities in the social media content. Although textual cyberbullying may contain obscenities and profanities, all the obscene text on social media may not be cyberbullying – considering Twitter especially, studies show that the rate of using offensive words is close to double on Twitter than in normal life (Steinmetz, 2014). Therefore, care must be taken in deciding whether a tweet constitutes cyberbullying or not, even if it contains obscenities. Prevalent approaches for the detection of cyberbullying are sequential in nature. A distributed paradigm is more suitable for the detection of cyberbullying in social media such as Twitter due to the below reasons:

1. Twitter data is generated in a distributed and asynchronous manner; it is better to detect the cyberbullying behavior at different locations in a network.

2. A sequential detection technique will suffer from bottlenecking and a single point failure.
3. A distributed detection can reduce the analysis time by exploiting the inherent parallel nature associated with the independent generation of tweets. (Mangaonkar, 2015)
4. The knowledge base used for detection is scattered across the network.

Applications such as Twitter© and Facebook® are inherently distributed, as their data gets generated in a geographically dispersed and an asynchronous manner. As data is getting created in parallel, these applications are bombarded with incoming data from various sources in a very rapid manner. In such a situation, the sequential approach is certainly a drawback. Hence, the processing technique applied to this data must not only be quick and efficient, but it also needs to be able to adapt to a distributed environment. Thus, if the cyberbullying detection has to be made online, the detection process must begin before the data reaches the central server.

This thesis proposes a distributed collaborative detection approach, that is, there will be many nodes in the network that run detection algorithms in parallel. These nodes may collaborate with each other if needed. A detection node may collaborate with other nodes because:

1. The other detection node may be better equipped at classification, and/or
2. The other node may be able to provide a second opinion about a particular tweet.

Figure 1 depicts the general idea regarding the proposed approach (Mangaonkar, 2015).

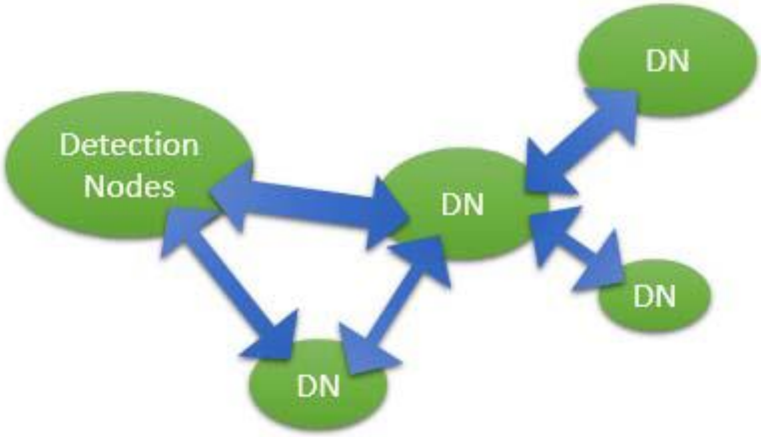


Figure 1-1: Distributed -Collaborative Cyberbullying detection approach

1.5 Objectives: Thesis Statement

The objectives of this thesis are listed below:

- To examine how a non-distributed, single server detection architecture suffers from bottleneck and single-point failures.
- To propose a distributed collaborative detection approach.
- To examine various distributed-collaborative techniques and their effect on the time and accuracy of detection of cyberbullying for a given Twitter dataset.
- To explore the possibility of the generalization of results obtained.

1.6 Evaluation

This thesis is considered a success if experiments performed with the distributed-collaborative approach are able to establish the following:

- Collaboration improves the performance of a detection node in network.
- Collaboration does not drastically reduce the performance of a detection node in network.
- It provides identical results on a different domain.

This work is tested mainly on a dataset created as part of this thesis. It assumes that the above approach will perform well in case of different data sets.

1.7 Contributions

- To propose a new distributed approach for the detection of cyberbullying in a Twitter dataset.
- To validate the need and effectiveness of the proposed approach.
- To examine the scalability and fault tolerance of the proposed approach to ensure it can be used in larger networks.
- To empirically validate the applicability of this approach in the detection of political tweets to illustrate its applicability in other domains.

1.8 Organization

This thesis is organized into six chapters. The first chapter, provides an introduction along with the motivation and problem definition. A literature review is presented in the second chapter. The third chapter discusses the proposed approach and experimental setup required. Additionally, it discusses the metrics used to evaluate the performance of the proposed approach. The fourth chapter provides the results of the experiments performed that test the need, effectiveness, and scalability of the proposed approach. The fifth chapter explores the possibility of the generalization of the proposed approach. Finally, chapter 6 provides the summary and conclusion.

2. BACKGROUND AND RELATED WORK

This chapter discusses the problem of cyberbullying, which, in the recent past, has emerged in social media. It provides an overview of proposed cyberbullying detection techniques in literature. Techniques that have been suggested up until now are either Lexicon-based or they use machine learning algorithms. Both these categories are discussed in this section.

2.1 Research Related to Cyberbullying Detection

A lot of research related to cyberbullying has been carried out in the fields of Psychology and Social Sciences (Dinakar, K., Reichart, R., & Lieberman, H., 2011). The psychological and emotional effects of cyberbullying are being extensively studied and analyzed (Hinduja ,S. & Patchin, J. W., 2007). Numerous surveys are carried out each year to better understand the effects and extent of the cyberbullying problem. These studies help to create guidelines that help the victims of cyberbullying deal with the problem. In some cases, advice is provided on how to protect oneself from cyberbullying. Traditional approaches focus more on increasing supervision by parents and teachers when children are using social media (Bosse, T., & Stam, S, 2011).

Although these approaches are very helpful, they are insufficient to address the problem of cyberbullying as a very small number of incidents are actually reported by children to their parents or teachers, or to a social media site ("Cyber bullying", 2017). Additionally, despite these measures discussed earlier, cyberbullying is increasing (Algar S. , 2017).

2.2 Content Filtering Software

There are content filtering software packages available on the market for Mobile and PC, including PureSight Multi (Online Child, 2011), PhoneSheriff (Worried about, 2017), CyberPatrol (CyberPatrol Parental, 2008), and WatchGuard XCS (Email and Web, 2014). These programs can be used by individuals or schools for the detection and prevention of cyberbullying. A few examples of these software are listed below:

Surfie by PureSight Multi: The PureSuite Multi-software package can be used by parents to supervise their child’s behavior on the Internet. A key feature of this application is that it supports parental monitoring of social media activity. It automatically monitors the Internet communication, filters verbal abuse and offensive content, and alerts parents if cyberbullying occurs. This award winning software contains Active Chat Inspector (ACI) and Active Content Recognition (ACR). It is one of the best parental control software. However, it does not provide support channels for victims. (Shipley R., 2017)

PhoneSheriff: This is parental control software for mobile control and tablets. It allows parents to monitor who their ward is in contact with. Its features include blocking phone numbers from calls and text messages from unwanted numbers, setting time restrictions on usage, blocking applications, and monitoring text messages. Additionally, it provides remote alerts and allows parents to track the location of a phone. (Worried about, 2017)

CyberPatrol Parental Controls: This parental control software can be used for both home and school use. It provides features such as filters for inappropriate Web sites, Internet access scheduler, controls programs, filtering of objectionable words from instant messages, and it monitors Web-surfing activity. However, it does not provide remote notification features. (CyberPatrol Parental, 2008)

WatchGuard (XCS): WatchGuard Extended Content Security (XCS) is a firewall that can be used by schools to prevent Cyberbullying behavior. It features the ability to block insulting, defaming comments and posts related to depression and suicide on social media sites (Stop Cyber-Bullying, 2011). WatchGuard is better than average help material, but it is tough to manage when custom settings need to be added, and it is less user friendly than some other software types. (WatchGuard Firebox , 2014)

Although, these software packages provide parental monitoring of online activities, they are limited to school yards and/or home. These programs act more as a firewall for cyberbullying, and hence, can be bypassed by savvy children. Additionally, parents often fail to fully utilize the potential of such software (Waugh, R. , 2014).

Further, there is no collaboration between different installations and instances of these packages. This results in localized detection of cyberbullying. Each installation and/or instance is limited by its own capacity and does not receive any help from other similar software packages/installations that are working towards the same goal.

2.3 Cyberbullying Detection Methods from Literature

Cyberbullying methods discussed in literature fall under two categories, lexicon based methods and machine learning based methods.

2.3.1 Lexicon based methods

These text classification methods of cyberbullying detection rely on a simple bag-of-words model. It creates a corpus of sensitive, abusive, and hateful words. Then it uses algorithms to look for these words in the online content that needs to be analyzed. Following are examples of the bag-of-words model: Chen et al. (Chen, Y., Zhou, Y., Zhu, S., and Xu, H., 2012) proposed a method called Lexical Syntactic Feature (LSF) for the detection of cyberbullying. It can be considered to be a smarter lexicon-based filtering method that considers the history of the negative behavior by the user on social media sites. It combines the message-level offensiveness detection with the user-level offensiveness detection to predict a user's potential to send offensive messages. For message-level offensive detection, this method heavily relies on BoW (Bag of Words), and the N-Gram techniques. User-level offensiveness calculations are performed by using the user's conversation history.

Kontostathis et al (Kontostathis, A., Reynolds, K., Garron, A., & Edwards, L, 2013) have examined specific words that are used by cyberbullies, as well as their context. These words are then used to form queries that were used to analyze cyberbullying content.

2.3.2 Machine leaning based methods

Text classification methods of cyberbullying detection use models created by machine learning algorithms. These models are created by training machine learning algorithms on datasets which are generally built by using manually labelled social media data. Additionally, they use some sort of preprocessing to make it more informative. Following are the studies that are from this category:

Yin et al. (Yin, D., Xue, Z., Hong, L., Davison, B. D., Kontostathis, A., & Edwards, L. , 2009) conducted cyberbullying detection experiments with a support vector machine classifier. They collected datasets from both chat-style websites and discussion-style websites, labelled them, and used it to train the classifier. In this study, they compared three different approaches N-Gram, Foul Language, and TF-IDF (Term Frequency-Inverse Document Frequency). The study showed considerable performance improvement when the basic TF-IDF approach was augmented with sentiment and contextual features.

Reynolds et al. (Reynolds, K., Kontostathis, A., & Edwards, L., 2011, December) used datasets from the 'Formspring.me' website, and then manually labelled and assigned weights to the post using Amazon's Mechanical Turk service that allows users to post tasks. For training algorithms, they used features of the post such as the number of bad words (NUM), density of bad words (NORM), and a weighted average of the bad words (SUM). Machine learning algorithms J48, JRIP, IBK, and SMO were used for this research.

The article by Dinakar et al. (Dinakar, K., Reichart, R., & Lieberman, H., 2011) discusses modeling the detection of textual cyberbullying. They explained that cyberbullying or harassment generally happens on the topics which are personal and/or sensitive to the victim. Race or culture, sexuality, intelligence, and physical attributes are the aspects that people cannot change about themselves. Hence, they are generally personal and sensitive. The model suggested by them (Dinakar, K., Reichart, R., & Lieberman, H., 2011) is based on the topic of sensitive classifiers. It breaks the complex problem of cyberbullying detection into less complex sub-problems. For each of the above mentioned topics for cyberbullying, classifiers are created using machine learning algorithms such as Naive Bayes, Support Vector Machine (SVM), etc. Authors then compare the binary classifiers and multiclass classifiers. A binary classifier classifies only for a single topic, and the multiclass classifier classifies for multiple topics. Their experiments suggest that the binary classifiers work better as compared to the multiclass classifiers. This model functions by sending a post to all binary classifiers to see if it is a cyberbullying post or not; this is done in a sequential manner. Dinakar et al. does not present any parallel processing for these classifiers.

Nahar et al. (Nahar, V., Li, X., & Pang, C., 2013) proposed a classification model for harmful posts detection using weighted TF-IDF and ways to identify predators and victims based on the number of harmful posts sent and received. To identify predators and victims, it uses a graph model that can also be used to check the level of cyberbullying victimization. The study focuses more on labeling the users depending on their behavior in cyberspace, thus allowing moderators to focus on groups of the most active predators and victims.

The cyberbullying detection proposed by Nahar et al. combines semantic features, second person pronouns, all other pronouns, and a dictionary of bad words from noswearing.com. It then applies a weighted TF-IDF scheme to improve the classification accuracy. For classification the LibSVM, implementation of an (SVM) has been employed. To address the issue of training the SVM with an imbalanced dataset, the oversampling of minority cases method is used.

An article (Tynes J. , 2014) published on LinkedIn© discusses benchmarking the precision and recall metrics of Twitter sentiment analysis. It uses machine learning-based approaches. Naive Bayes and a voter algorithm are trained on a manually labelled dataset of 3,424 tweets. Another algorithm, maximum entropy, is trained using a different labelled dataset with 1.6 million tweets. The results show that MaxEnt is the best-performing algorithm in terms of precision and recall.

Research by Kasture (Kasture, A. S. , 2015) takes the benefit of a psychometric evaluation tool, Linguistic Inquiry and Word Count (LIWC). This tool helped in creating word categories using the Twitter dataset of 1313 unique tweets. Additionally, the tool evaluates frequencies with which these word-categories are used. This text categorization and word-usage quantified by the tool was used to create a training dataset. The author trained a variety of Weka machine learning algorithms using preprocessed dataset to create a predictive model of cyberbullying detection. This research then analyzes the performance of various Weka machine learning algorithms. This research adds psychometric evaluation to the original detection problem.

Zhang, X et al. (Zhang, X., Tong, J., Vishwamitra, N., Whittaker, E., Mazer, J. P., Kowalski, R., Hu, H., Luo, F., Macbeth, J. & Dillon, E, 2016) address the issue of noise and errors in social media data, which is one of the challenges faced by Cyberbullying detection algorithms. The study uses Twitter and Formspring datasets. To address the challenge of misspelled words, Zhang, X et al suggest Word-to-Pronunciation conversion using speech synthesizer software and generation of phonetic representation for each word. This study uses the following machine learning algorithms: SVM, Multilayer Perceptron, J48 Decision Tree, CNN pre-trained, CNN Random, and PCNN. PCNN method developed by Zhang, X et al. outperforms all models on all metrics compared to previous work by Kasture.

All the proposed work discussed above is focusing on developing a single efficient and effective method for the detection of cyberbullying. The ways in which cyberbullying happens are continuously evolving (McQuade, S. C., Colt, J. P., & Meyer, N. B, 2009). A method that is effective on a certain dataset, may not be as efficient on others. This requires collaboration between different methods of cyberbullying detection.

In this research, we have used Machine Learning-based approaches for cyberbullying detection. However, we do not dictate the type of cyberbullying method that should be used by the detection nodes in the network. Machine Learning-based approaches are used mainly because of the popularity and effectiveness of them.

3. METHODOLOGY AND EXPERIMENTAL SETUP

In this chapter, our proposed approach for cyberbullying network is discussed. The chapter further discusses, performance metrics, parameters under study, datasets, and algorithm used.

3.1 Distributed-Collaborative Detection Approach.

The proposed system employs a collaborative approach to classify a tweet as “bullying” or “non-bullying.” The architecture of the proposed system is shown in Figure 3-1. Each server shown in this figure is a detection node. A detection node is an independent entity that has an ability to receive an incoming tweet from a clients and classify them as a bullying or a non-bullying tweet. This distributed-collaborative detection approach proposes that these detection nodes work in concert to accurately classify tweets as either bullying or non-bullying communication.

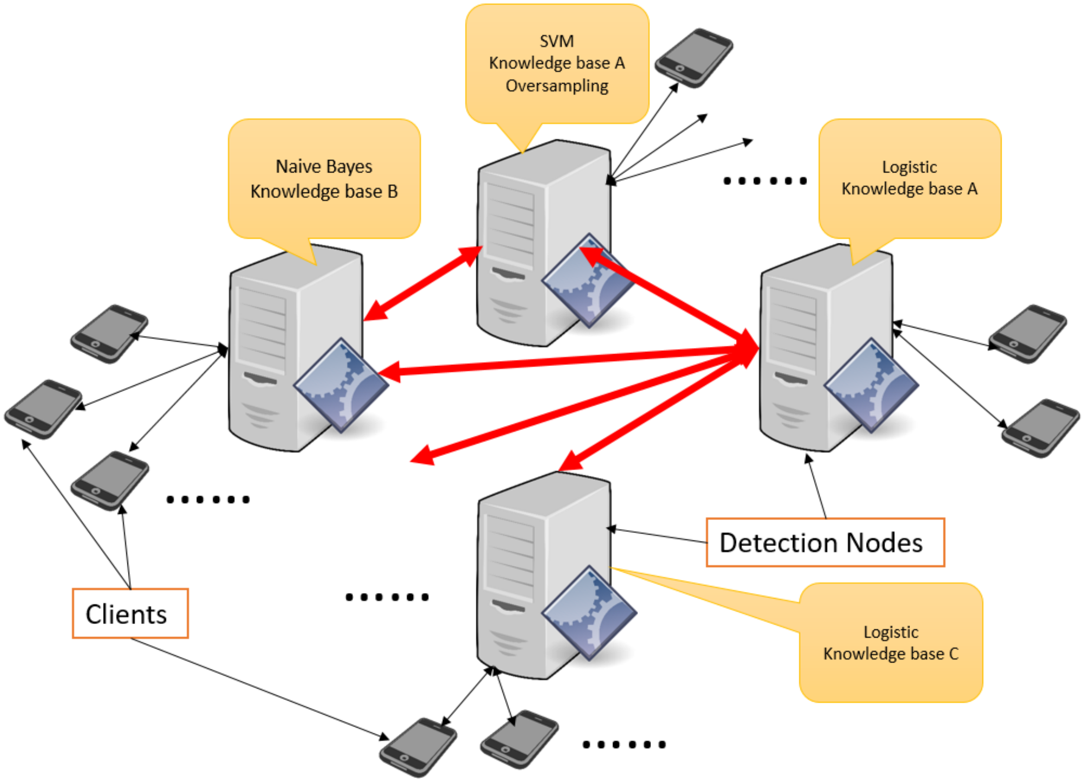


Figure 3-1: Architecture of distributed-collaborative cyberbullying system

Prevalent approaches such as LSF (Chen, Y., Zhou, Y., Zhu, S., and Xu, H., 2012), PCNN (Zhang, X., Tong, J., Vishwamitra, N., Whittaker, E., Mazer, J. P., Kowalski, R., Hu, H., Luo, F., Macbeth, J. & Dillon, E, 2016) for detecting cyberbullying are sequential in nature. Applications such as Twitter© and Facebook® are inherently distributed as their data gets generated in a geographically dispersed and asynchronous manner. As data is created simultaneously and continuously, these applications are constantly flooded with incoming data from various sources. In such situations, the sequential or linear approach is unable to manage the influx of messages and a bottleneck situation is created. Hence, any processing technique applied to this data must not only be rapid and efficient, but the process should also be adaptable in a distributed environment. Thus, if the cyberbullying detection must occur online, the detection process should commence before the data reaches the central server to reduce analysis time and to avoid bottleneck. This requires the detection mechanism to work in distributed mode, that is, many nodes or detection points are required in the network that will perform detection algorithms in parallel as suggested in our previous work (Mangaonkar, 2015).

Additionally, over time each detection node can build a different and distinct knowledge base by collecting feedback from users and use it to update the classification model. As no two tweets are exactly alike, each detection node learns, and thereby increases its interpretation and bullying detection capability based on previous messages. Further, each detection node implements a different algorithm for classification designed to classify particular patterns of tweets. With individual classifying processes, in order to ensure the most effective classification process, collaboration between detection nodes is vital. A detection node may collaborate with other nodes because:

1. The other detection node may be better equipped at classification, and/or
2. The other node may be able to provide a second opinion about a particular tweet.

3.2 Performance Metrics

Precision, and recall (Tynes J. , 2014) are the performance metrics used to examine and compare the performance of various classification techniques. This study also use time as a metric to analyze the effect of collaboration on classification time. The accuracy is provided for reference only. These metrics are discussed below.

1. Accuracy: This metric measures the number of tweets correctly classified. It can be calculated using the following formula:

$$\text{Accuracy} = \frac{Number\ of\ True\ Positive(TP) + Number\ of\ True\ Negavtive(TN)}{Total\ number\ of\ tweets} \quad (1)$$

2. Precision: This metric measures the number of tweets classified by the algorithm as bullying and actually prove to be bullying tweets. Precision can be calculated using the following formula:

$$\text{Precision} = \frac{Number\ of\ True\ Positive\ (TP)}{Number\ of\ True\ Positive\ (TP) + Number\ of\ False\ Positive(FP)} \quad (2)$$

3. Recall: This metric measures how many bullying tweets are actually detected by the algorithm. Recall is calculated by the following formula:

$$\text{Recall} = \frac{Number\ of\ True\ Positive\ (TP)}{Number\ of\ True\ Positive\ (TP) + \ Number\ of\ False\ Negavtive\ (FN)} \quad (3)$$

4. Time: This metric measures the time taken by the algorithm for classifying a given number of tweets.

The training set/test set used in this experimental study will always have a lower percentage of ‘bullying’ tweets than ‘non-bullying’ tweets to imitate real life scenario. For this scenario, ‘Accuracy’ is not a correct measure of performance. For example, consider a test set with 80% of ‘non-bullying’ tweets, if all the tweets get classified by a cyberbullying detection technique as ‘non-bullying’ then according to the equation mentioned in this section, the pattern has 80% Accuracy. However, it was not able to catch a single ‘bullying’ tweet. Thus, ‘Accuracy’ is not considered as an important measure in this research.

Additionally, even if the cyberbullying detection technique discovers one percent of tweets that can be classified as bullying accurately, increasing to 100 percent precision, that pattern is not useful as most of the ‘bullying’ tweets are overlooked. Hence, recall is the essential measure as it is more important to detect more cyberbullying posts at the cost of false positives than it is to detect a small number of cyberbullying tweets accurately (Zhang, X., Tong, J., Vishwamitra, N., Whittaker, E., Mazer, J. P., Kowalski, R., Hu, H., Luo, F., Macbeth, J. & Dillon, E, 2016). However, these false positive should have a limit. Therefore, our approach considers recall, precision, and time to analyze and compare performance of different distributed-collaborative patterns.

3.3 Parameters in Distributed-Collaborative Detection Approach

This section discusses the parameters that impact the performance of different distributed-collaborative configurations. These parameters affect the performance metrics indicated in the previous section. By fixing and varying values for these parameters, different experiments are indicated in Chapter 4 for evaluating the performance of distributed-collaborative configurations.

3.3.1 Nodes in the network

The number of nodes in the network is the basic parameter that identifies how many detection nodes exist in a network that either functions as an entry point into the proposed system or provides an opinion on a given tweet.

3.3.2 Training set associated with each node

Each node’s Training Set, (made up of tweets) along with the classification algorithm, combine to create various models used for the classification of test tweets by the detection nodes. A model (and hence, in turn the detection node using it) performs better with a more effective training set. A more effective training set is one that provides adequate knowledge and draws information from actual examples to build accurate models. A “complete” training set is the one that builds a model with the capacity to classify every single tweet correctly. In actuality, a “complete” training set may not be present on a single node due to the following reasons, including:

- an influx of new knowledge through the tweets it is classifying;
- new ways to bully being developed every day; and
- the training set updates are performed at different intervals on different detection nodes.

In essence, each detection node may have a portion of the “complete” training set required for classifying tweets. The part of this “complete” training set (or knowledge base) included with a detection node will define its capacity to classify a given test tweet. Classification performance of a specific detection node on a certain test set will vary depending on the node’s knowledge base when the same classifying algorithm is used. In this case, collaboration may allow combining performances of detection nodes for better classification. Hence, it is important to consider a distinct training set or knowledge base present on each node when performing experiments for the distributed-collaborative approach.

3.3.3 Entry point(s)

Entry point is a detection node that is first contacted by the client for the classification of a tweet. If the node has a poor detection threshold, collaboration is employed to increase classification. This study assumes that, when a detection node classifies a tweet as ‘non-bullying,’ it may be classifying it wrong due to its inability to detect the bullying content in it. Additionally, collaboration may not affect the performance of better performing nodes. Thus, to analyze the use of any collaboration configuration, it is important to take into consideration the entry point into the distributed-collaborative system. Additionally, there can be more than one entry point concurrently working. It is important to see the impact on classification time in this case.

3.3.4 Algorithms

Various machine learning algorithms (as indicated in chapter 2) are used in literature for the text classification problem. Among these algorithms Maximum Entropy, Naive Bays, and Support Vector Machine were employed to classify a tweets in our previous work (Mangaonkar, 2015) . Results of this work establish that Maximum Entropy (i.e. Weka Logistics) classification algorithm performs better as compared to Naive Bayes and SVM on cyberbullying twitter dataset.

It is possible to assign different machine learning algorithms to different detection nodes within the same network. The machine learning algorithm used by a detection node will define the classification capability and the learning capability of each node. Hence, overall performance of distributed-collaborative configuration will be affected by the algorithms used on the detection nodes.

3.3.5 Number of opinions

The opinions that are provided by other detection nodes to the entry detection node are central to a distributed-collaborative system. The entry point has to invest time and resources for each opinion that it is receiving. The most accurate approach is to seek opinion from all the available nodes in the network. While this approach may give an excellent performance, classification time under this approach will be high.

3.3.6 Selecting detection node

The nodes providing the opinion to the entry point affect the quality of final classification. While the assessment taken from a more effective performing node could improve the performance of the detection, an assessment taken from a poor performing node might bring down its performance. In short, selecting a node to extrapolate an assessment from it will be a defining factor in the performance of the detection node.

3.3.7 When to collaborate

As discussed in Section 3.1, the detection node can seek an opinion from other nodes for two reasons; first, other detection nodes may be better at classification, and/or, second, to get a second opinion about a particular tweet. Taking a valuation from other nodes does involve extra processing and hence extra time. A Distributed-Collaborative system may decide to collaborate for every tweet or when the need arises to take a second estimation. Classification time will increase or decrease as more or less collaboration is being performed in the distributed-collaborative approach.

3.3.8 Result merging technique

The results merging techniques essentially combine results of different classifiers. These classifiers are either using different training sets, and/or different learning methods. This is similar to using ensembles of classifiers concept where results of classifiers are combined to improve performance of individual classifier (Ensembles of classifiers, 2016). There are three opinion merging techniques that are primarily used in this research.

OR Merging: When more than one assessment is collected at the entry point, any assessment that classifies the tweet as ‘bullying’ produces a resultant classification by the entry point as ‘bullying’.

AND merging: This is used when more than one assessment is collected by the detection node; if all other assessments classify the tweet as ‘bullying’ then the resultant classification by the entry point is ‘bullying’.

Majority merging: When more than one opinions are collected, if more than half of the assessments classify the tweet as ‘bullying’ then the resultant classification by the entry point is ‘bullying’.

OR merging increases the recall at the cost of more false positives; AND merging increases precision at the cost of more false negatives; Majority merging covers for fewer poor performing nodes in the system by overriding their classification. Performance of the detection node varies with the type of merging technique it is using. Thus, it is an important parameter to look at in distributed-collaborative experiments.

3.4 Twitter Data Collection

Twitter data used in this study was collected by downloading the file that was shared by Li, Rui. As Dataset: UDI-TwitterCrawl- Aug2012 (Li, 2012). Twitter data had been collected through crawling, which contains 3 million user profiles and 50 million tweets. The dataset was collected in May 2011 and it was split into 1,417 files. This data was used to create a consolidated file with 200,000 tweet messages which was further used in experiments performed for Distributed Detection to Avoid Bottleneck section.

Additionally, in order to analyze the performance of various distributed collaborative configurations, 6,257 tweets were collected including 1,847 as bullying tweets and 4,409 as non-bullying tweets. To build this dataset, an extensive study was performed on the Internet; multiple bullying instances were studied to collect the bullying tweets. Further, the batch of tweet was collected from Twitter directly and non-bullying tweets were tagged manually after studying each tweet.

This initial dataset was then split to create a training dataset and a test dataset. The test dataset that was taken out from the original dataset had the following composition:

Total tweets: 626

Bullying tweets: 180

Non-bullying tweets: 446

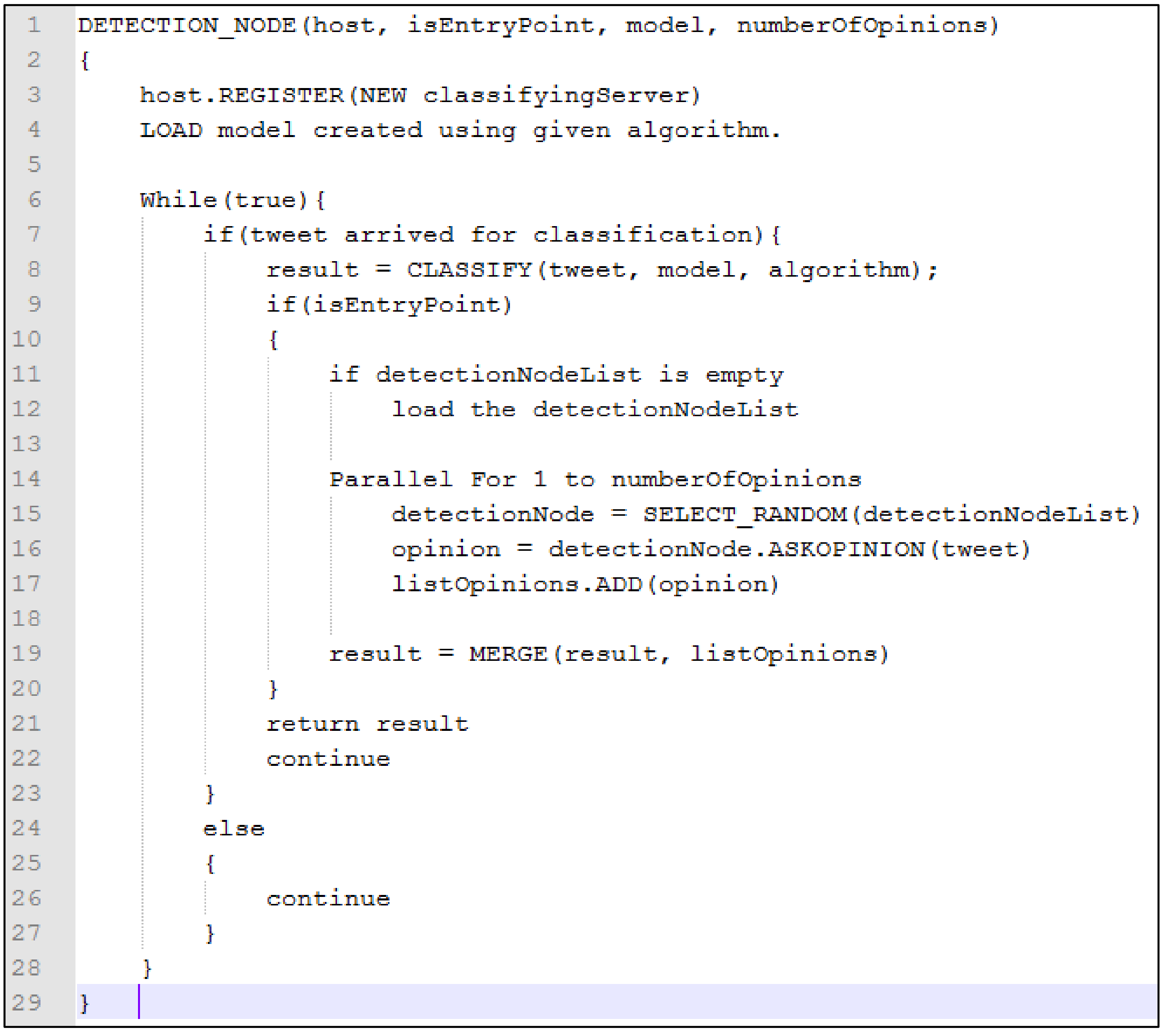
Remaining tweets were part of the training dataset. These tweets were used for training Weka (Waikato, n.d.) machine learning algorithms such as Maximum Entropy (Weka Logistic), Naive Bayes, and SVM.

3.5 Model Creation with Weka Algorithms

Weka machine learning algorithms that were discussed in our past work “Collaborative Detection of Cyberbullying Behavior” (Mangaonkar, 2015) were used in this study. Weka machine learning algorithms support different classification algorithms such as Support Vector Machine, Naive Bayes, and Maximum Entropy (Logistics). These machine learning algorithms (due to their effectiveness as indicated in our past work) were trained to create models that were used for the classification of cyberbullying tweets. These algorithms were trained using supervised learning techniques that infer a classification function from a labeled training dataset created as indicated in Section Twitter Data Collection. Weka provides filters that were used to extract word vectors from the training tweets. A word vector fine tuning parameter Minimum Term Frequency for a word allows filtering out words whose frequency in a training dataset is below the expected value, which was set to 2. This value was selected as the result of past work (Mangaonkar, 2015), which suggests that it provides balanced precision and recall results. The other fine tuning parameter, ‘Tokenizer’ that identifies phrases or sequences of words that always indicate bullying was set to NGram. These values were determined from the experiments performed on the dataset. We also set the cross validation to 10 folds. Cross-validation is a model validation technique; it measures how the models created using a statistical analysis will generalize to an independent data set (Cross-validation , 2017). Therefore, to ensure better prediction performance on an independent dataset, we have used 10 fold cross-validation to validate models generated using Weka.

3.6 Pseudo Algorithm for Detection Node

The detection nodes have the ability to classify a tweet as a bullying or non-bullying tweet based on the training data and a Weka classification algorithm that it has. The pseudo code for a detection node is given in the figure below.



```
DETECTION_NODE(host, isEntryPoint, model, numberOfOpinions)
{
    host.REGISTER(NEW classifyingServer)
    LOAD model created using given algorithm.

    While(true){
        if(tweet arrived for classification){
            result = CLASSIFY(tweet, model, algorithm);
            if(isEntryPoint)
            {
                if detectionNodeList is empty
                    load the detectionNodeList

                Parallel For 1 to numberOfOpinions
                    detectionNode = SELECT_RANDOM(detectionNodeList)
                    opinion = detectionNode.ASKOPINION(tweet)
                    listOpinions.ADD(opinion)

                result = MERGE(result, listOpinions)
            }
            return result
            continue
        }
        else
        {
            continue
        }
    }
}
```

Figure 3-2: Pseudo-Algorithm for detection node.

Each detection node accepts four parameters. The first parameter is the server name or host name on which it is located. The second parameter is a Boolean; when true, the detection node is receiving tweets from the client. The third parameter is a model created using a specific machine learning algorithm. This model makes predictions regarding test tweets.

The last parameter decides how many opinions a detection node is going to seek to make a final classification decision. As indicated in figure 3-2, the steps of the algorithm are:

1. Create a new tweet classifying the server and register it on the machine.
2. Load the knowledge base and set the algorithm used for classification.
3. Set the node selection type and opinion condition.
4. Wait for Tweets to arrive.
5. When a tweet arrives, classify the tweet using knowledge base and algorithm.
6. Check if the current detection node is an Entry Point.
7. If it is an entry point, check if taking opinions is expected in the current case.
8. If opinions are to be taken, then load detection nodes to take opinion from. For each opinion to be taken, select a detection node from the list of detection nodes using the random selection and ask opinion. Merge all the opinions using the merge technique.
9. Return classification result.

This concludes discussion about experimental setup. The proposed approach discussed in this chapter, along with parameters, datasets, and detection node algorithm are used for experiments that are discussed in the next chapter.

4. EXPERIMENTAL WORK

This chapter discusses multiple experiments that were carried out in order to empirically validate the proposed distribution-collaborative approach to perform cyberbullying detection in online social media networks such as Twitter.

4.1 Distributed Detection to Avoid Bottleneck

As of today, the average number of tweets observed per second is 6,000. On August 3, 2013, Twitter observed (Twitter Usage Statistics) twenty times (143,199) heavier traffic than average. It is important to be prepared for the growth of the Internet and heavier tweet generation rates when cyberbullying detection is being implemented in an online environment. In order to validate the hypothesis that the sequential approach to detect cyberbullying in an online environment will overwhelm a single classification server by creating a bottleneck, we carried out the following analysis.

4.1.1 Theoretical analysis

The server response time (T_R) can be represented by equation 1. Where $T_{transmission}$ is the time to transmit a tweet to the detection node, T_{wait} is the time the tweet is at the detection node before it is classified, and $T_{classify}$ is the time required to classify a tweet.

$$T_R = T_{trasmission} + T_{wait} + T_{Classify} \qquad (1)$$

Tweets have a defined length of 140 characters. Therefore, the classification time for each tweet does not vary drastically and can be considered a constant value. Thus, if there is no wait time, the response time will be a function of transmission time. The response time equation can be rewritten as equation 2, where C is the constant classification time.

$$T_R = T_{trasmission} + C \quad when\, T_{wait} = 0 \qquad (2)$$

However, if the tweets are being sent by a client program at the detection node at a much faster rate than the constant classification time (C), a queue will start building at the detection node. When the interval between two tweets is very short, the T_{wait} is much higher than the C and $T_{trasmission}$, thus, T_R will become function of T_{wait}. Consequently, the T_R can be rewritten as equation 3.

$$T_R = T_{wait} \quad when\ T_{wait} \gg T_{trasmission} \tag{3}$$

Essentially, as the arrival rate of tweets at the detection node increases, and the time interval between two tweets decreases, the average response time for a tweet will increase. This will create a bottleneck at the detection node when higher tweet arrival rates are present.

4.1.2 Experimental analysis

The pseudo algorithm shown in Figure 3-2 was used for implementing a detection node. Preprocessing and classification time as 1 millisecond was considered during this analysis. A consolidated file including 200,000 tweets was used as an input to the experimental study (Li, 2012). Further, a client application was developed with the capability of reading a tweet and transferring it to the detection node at a given interval. In this experimental study, T_R for each tweet was calculated by varying the delay between tweets. The average T_R was calculated in order to normalize the response time for each tweet. Figure 4-1 shows the average T_R with respect to the delay between tweets.

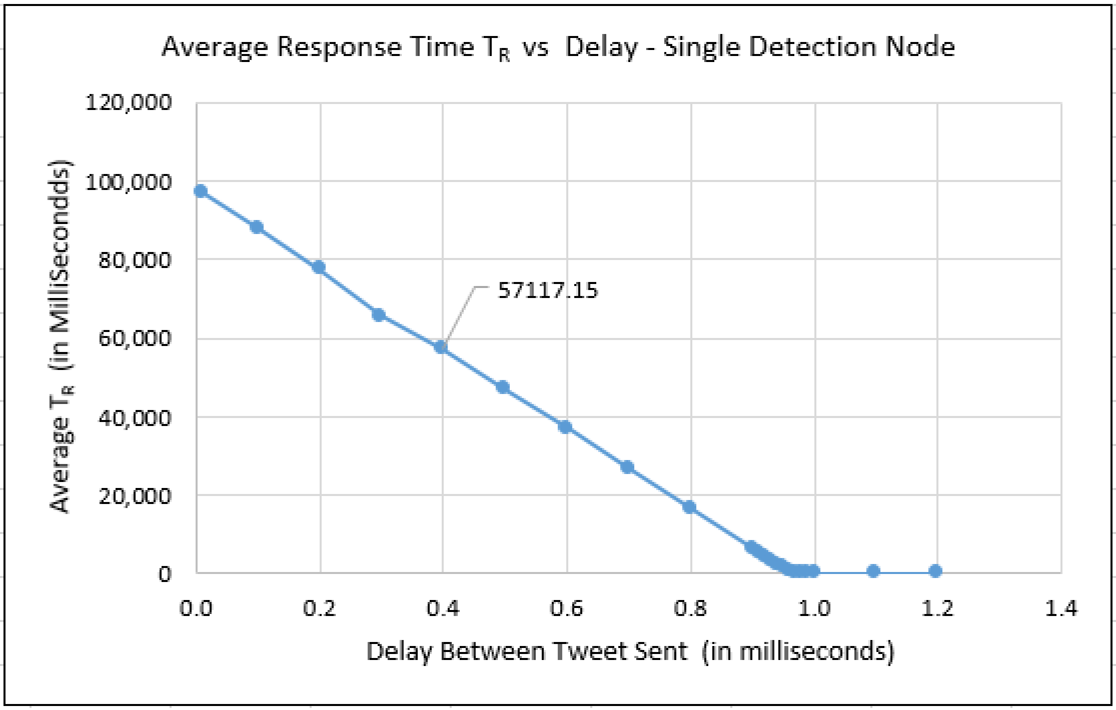


Figure 4-1: Relationship between Average TR (Response time) with respect to the delay between tweets at a single detection node condition.

As seen in Figure 4-1, when the delay is 0.4 milliseconds, the average T_R is around 60,000 milliseconds. When the delay is around 1.2 milliseconds, T_{wait} is very low approaching 0. From this experiment, it has been concluded that a higher arrival rate of tweet causes T_{wait} at the detection node to increase by creating a large queue, ultimately causing a bottleneck created at the detection node.

In order to understand the impact of the distributed approach on the T_R, these experiments were repeated with two and three detection nodes. The client application was modified to send tweets at a regular time interval to a detection node selected at random from existing nodes in the network. The average T_R was recorded in the same manner as single detection node study. It is expected that as higher numbers of detection nodes are present in the network, and the tweets are distributed among them, it will reduce the tweet arrival rate at individual detection nodes.

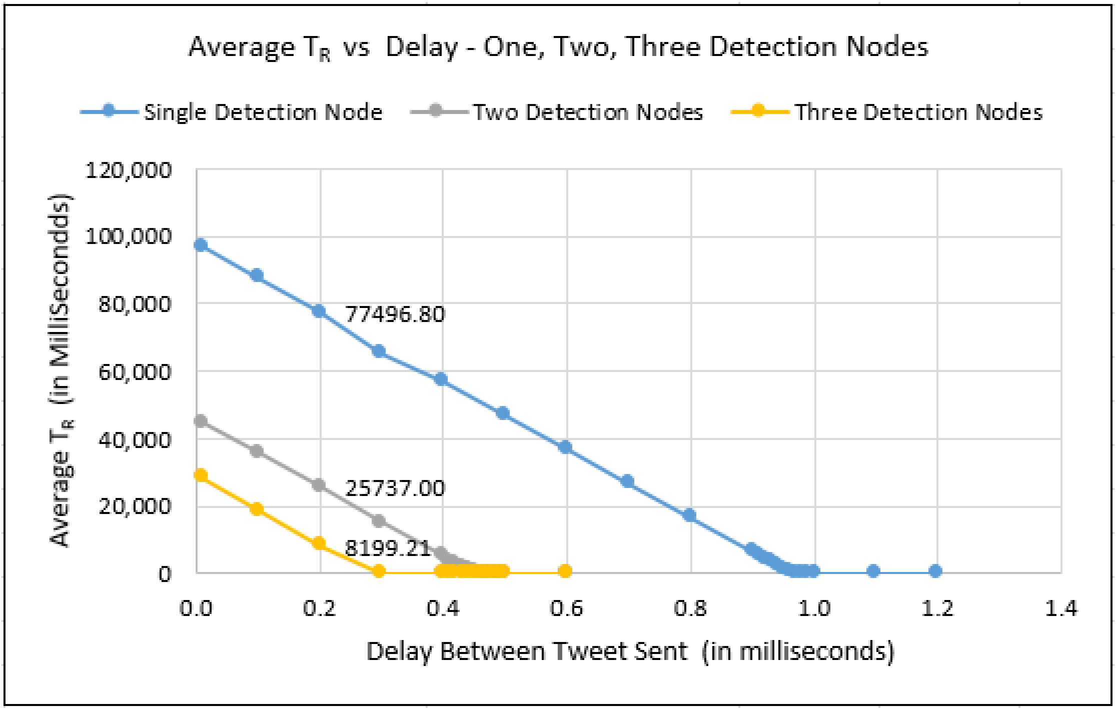


Figure 4-2: Average T_R with respect to the delay between tweets for single, double and triple detection nodes

It can be observed from Figure 4-2 that when the delay between two tweets is 0.2 milliseconds, for a single detection node, the average T_R is close to 80,000 milliseconds. For two and three detection nodes the average T_R value is close to 25,000 milliseconds and 8,000 milliseconds respectively. It can be clearly seen from the graph that if there is only a single detection present in the network, it gets quickly overwhelmed. However, when two detection nodes are present in the network, they do not start building queues until the interval between two tweets is half of the classification time. For three nodes, queue building does not start until the interval between two tweets is one third of the classification time. In essence, as the number of detection nodes in the network increases, the response time remains constant and is mainly governed by the time required by the classification algorithm. The number of other tweets getting generated has lesser impact on it.

Hence, we conclude that if a single detection node is used for cyberbullying detection in an online environment, it will create a bottleneck as tweets start getting generated at a higher rate than classification time. To provide classification in constant time in an online environment, it is necessary to use the distributed detection approach.

4.2 Distributed-Collaborative Detection to Unify Scattered Knowledge Base

In this section, the experiments performed for validating the need for a distributed-collaborative approach are discussed. A distributed-collaborative approach is required when the entire knowledge base is not present at the single-detection node. These experiments use recall, precision, and time performance metrics. Accuracy is provided only for reference.

4.2.1 Experimental configuration for three detection nodes

For these experiments, the training set discussed in Section 3.2.2 was used to create three disjoint training sets by keeping the ratio of bullying over non-bullying tweets the same as the original training set. These newly created training sets were used for training three detection nodes (DN) with the logistics algorithm. For the experiments in this chapter, this research has used Logistic algorithm as our previous work (Mangaonkar, 2015) and work by Tynes et al. (Tynes J. , 2014) established that it performs better on a Twitter dataset. This approach created a scenario where each detection node will hold part of the entire knowledge base.

In order to study the performance of each detection node under this approach, the test set was classified by each detection node individually. The classification performance of each node is compiled in Table 4-1 in terms of accuracy, precision, recall, and time. Additionally, the entire training set is used for training a single detection node. This node was then used for classifying test set tweets, and results were provided in the same table as a reference.

Once the training set is split, each of the detection nodes is trained with fewer examples of ‘bullying’ and ‘non-bullying’ tweets. For some detection nodes, the training set does not have enough good examples of tweets. Thus, these individual detection nodes will have inferior performance compared to the reference which is evident by lower precision and recall values as shown in Table 4-1. It was also observed that the performance of DN2 is worse than DN1 and DN3. Degradation in the performance of individual DNs is attributed to the incomplete knowledge set, further emphasizing the need for collaboration between these detection nodes in order to take advantage of the entire knowledge base.

Table 4-1: Performance of classification algorithm at individual detection nodes with split knowledge base compared with reference model with full knowledge base – Three node network

Node	Accuracy (Percentage)	Precision (Percentage)	Recall (Percentage)	Time (seconds)
Detection Node 1 (DN1)	64.69	44.33	97.17	0.67
Detection Node 2 (DN2)	71.40	25	0.56	0.60
Detection Node 3 (DN3)	65.33	44.76	96.61	0.56
Reference	66.29	45.50	97.17	0.64

During this research the reason for the decrease in performance of the DN2 was investigated. The study performed by Ng et al. (Ng, 2002) provides a valuable insight that when dataset size is small, the logistics regression does not perform in some cases as it does not approach its asymptotic error. In our study, training dataset used for DN2 had one similar decrease in performance due to smaller sample size.

Collaboration Cases: Description, Execution, and Results

The three detection nodes, DN1, DN2, and DN3 were used to form a network, experimentally. The performance of collaborative configurations among these nodes was evaluated. It was hypothesized that the distributed-collaborative approach would increase the performance of a poor performing node without drastically reducing performance of the better performing nodes. In order to experimentally validate this hypothesis, multiple experimental configurations, also called cases, were created by changing values of detection node algorithms. For each case, the experiment was repeated five times to validate the results. In order to normalize the results, average accuracy, precision, recall, and time were calculated.

The parameters that were defined, and are identical from Case 1 through Case 7, are listed in Table 4-2.

Table 4-2: Fixed parameter values for Case 1 to Case 7 – Three node network

Parameter Name	Parameter Value
host	DN2
isEntryPoint	True for DN2
Model	D/3 that is not performing better on current test set
Algorithm	Logistics
detectionNodeList	DN1 and DN3
selectionType	Random

The entry point for all cases was selected as DN2 (the least performing detection node) to observe any improvement due to the distributed-collaborative approach. All three nodes have training set and use the Logistics algorithm to create a classification model. The client application has been modified such that it will send a tweet only to DN2 for classification. DN2 takes the opinion of DN1 and/or DN3 nodes regarding classification of the given tweet. DN2 will randomly select the node to take opinion from.

Collaboration parameters such as nuberOfOpinions (number of opinions taken by detection node), opinionCondition (when does a node initiates collaboration), MERGE (technique used to merge opinions) were varied to create cases 1 to 7. Details of these cases and parameter values were compiled in Table 4-3. The goal of the experimental study is to evaluate the correlation between each parameter with the performance of the detection node. Understanding this relationship will be valuable to create optimal collaboration patterns that will provide better precision and recall without adding extra cost in terms of time.

Table 4-3: Design of experiments by varying collaboration parameter values – Three node network

Cases	numberOfOpinions	opinionCondition	MERGE
Case 1 (C1)	1	True i.e. Always	OR
Case 2 (C2)	2	Always	AND
Case 3 (C3)	2	Always	OR
Case 4 (C4)	2	Always	Majority
Case 5 (C5)	2	Classified as non-bullying	AND
Case 6 (C6)	1	Classified as non-bullying	OR
Case 7 (C7)	2	Classified as non-bullying	OR

In the following discussion, each experimental case is discussed in detail. Additionally, for the first couple of cases we have provided tweet level analysis of results received for the test set.

Case 1:

In this case, DN2 will randomly seek an opinion from either of the other two DNs every time. 'OR' merging technique was utilized, which will classify the tweet as 'bullying' if either detection node classifies the tweet as 'bullying.' It is expected that the OR merging technique will increase the recall at the cost of the precision. The results of this experiment are compiled in the Table 4-4:

Table 4-4: Performance of DN2 in Case 1 with respect to no-collaboration case – Three node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.40	25	0.56	0.60
Case 1	64.53	44.22	97.28	2.12

The recall of the DN2 is increased due to collaboration, and it is almost equal to the better performing nodes. This is due to the fact that both other nodes are overriding wrong classification done by the DN2. As both the other nodes have better recall, when DN2 wrongly classifies a tweet as 'non-bullying' and asks for an opinion from DN1 or DN3, they provide correct classification as 'bullying' and due to OR merging the tweet is considered as 'bullying.' Additionally, precision is increased as a result of the higher number of true positives and opinions that are taken from nodes that have precision greater than 25%. Hence, as expected, performance improvement is achieved in this case using collaboration. A detailed analysis at the tweet level is provided below for an individual run: In case of no-collaboration, DN2 classifies only four tweets as bullying and out of these four, only one is a true positive, and the other three are false positives. In the case of collaboration, only 4 times DN2 classifies tweet as bullying, the rest of the time it is asking opinions from either DN1 or DN3. Collaboration with OR merging increases true positives as an assessment from a better performing node is ORed with DN2's assessment. As both DN1 and DN3 has recall close to 96%-97%, the resultant recall in this case is close to 97%. However, it increases false positives at the same time. DN1 has 216 false positives (precision 44.3%) and DN3 has 211 false positive (precision 44.8%) tweets. This propagates in the collaboration results which has 218 false positives (precision is 44%).

Case 2:

Variable parameters used in case 2 are indicated in Table 4-3. The difference between case 1 and case 2 was the approach in merging technique; 'AND' merging was used in case 2 in place of 'OR' merging. In 'AND' merging, the tweet is considered as 'bullying' if either entry point classifies it as 'bullying' or all the other opinions collected have classified the tweet as 'bullying.' It is expected in this case that the precision will be increased and recall will be decreased for the entry detection node.

Table 4-5: Performance of DN2 in Case 2 with respect to no-collaboration case – Three node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.40	25	0.56	0.60
Case 2	65.81	45.12	96.61	2.10

The result of this experiment is seen in the Table 4-5. The precision for DN2 was increased. It improved from 25% in the no-collaboration case to 45% in collaboration. The 'AND' merging technique caused any instance that was incorrectly classified as 'bullying' by any one detection node in the network was corrected by other detection nodes, eventually increasing the precision . However, if any one node correctly classifies a tweet as 'bullying', it is overridden by a wrong classification from other node. The other nodes in the network are better performing, hence, the ‘AND’ operation on their results is producing correct classification. A detailed analysis at the tweet level for an individual run is provided below: In case of no-collaboration, out of 177 bullying tweets from the test set only one true positive was detected, and 176 bullying tweets were false negatives. In case of collaboration, there are only 4 times DN2 classifies tweet as bullying, the rest of the time it is asking opinions from DN1 and DN3. Collaboration adds true positives from DN1 and DN3 and hence, recall improves. Due to the use of the ‘AND’ operation, false negatives from DN3 propagate in the final result, limiting recall of the collaboration to recall of DN3. In the case of no-collaboration, DN2 classifies only four tweets as bullying and out of these four, only one is a true positive, and other three are false positives. In the case of collaboration, only 4 times DN2 classifies a tweet as bullying, the rest of the time it is asking opinions from DN1 and DN3. Collaboration adds true positives where both DN1 and DN3 has classified a tweet as bullying. Additionally, due to use of AND operation, false positives from DN1 are corrected by DN3 and false positives from DN3 are corrected by DN1. There are a total 17 individual false positives that are corrected after AND merging. This increases precision of collaboration. Both DN1 and DN3 has precision 44.33 and 44.76 respectively. ‘AND’ merging increases collaboration to 45.12 percent.

Case 3:

In this case, the parameter ‘numberofopinions’ was set as ‘2’ and ‘OR’ merging technique was used. i.e., ‘numberOfOpinions’ was varied from case 1 and it was set to ‘2’. It is expected that the recall will be increased similar to Case 1. The intention behind this case was to understand the impact on the parameter ‘time’ when the detection node was seeking opinions from multiple other nodes.

Table 4-6: Performance of DN2 in Case 3 with respect to no-collaboration case – Three node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.40	25	0.56	0.60
Case 3	63.73	43.69	97.74	2.00

Results obtained by executing Case 3 are compiled in Table 4-6. As per the results, the recall and precision are improved since both the other nodes are performing better.

Similar to Case 1, collaboration with OR merging increases true positives as assessment from a better performing nodes are ORed with DN2's assessment. False positives in this case are the union of false positives from all three nodes which is 223. This results in lower precision than Case 1. The average time taken to ask two opinions was not significantly different than the time taken to ask single opinion, i.e., case 1. This is due to the fact that these two opinions are asked in parallel by using threads.

Case 4:

In Case 4, 'Majority' merging technique was used while all other parameters were identical to of the Case 3. In this merging approach, a tweet was classified as 'bullying' only when more than half of the detection nodes classify a tweet as 'bullying'. All variable parameters used in Case 4 are compiled in Table 4.3. Incorrect classifications of the DN2 were overridden if the remaining two detection nodes were accurately classifying the tweet unanimously, eventually improving the performance of the DN2.

Table 4-7: Performance of DN2 in Case 4 with respect to no-collaboration case – Three node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.40	25	0.56	0.60
Case 4	65.97	45.24	96.61	2.10

Results by executing case 4 using client application were compiled in Table 4-7. It can be seen from results that both recall and precision of the DN2 were improved. The DN1 and DN3 have better recall (that is 2 out 3 assessments provide higher true positives); therefore, resultant assessments in the Majority collaboration have higher true positives. Additionally, Majority merging false positives from a single node are corrected by other detection nodes. Thus, in this case, Majority merging helps in keeping false positives in control while improving the number of true positives. There are 207 false positives reported in this case of Majority merging which less as compared to AND or OR merging.

Case 5:

In Case 5, 'AND' merging technique was used while all other parameters were identical to Case 2. The variable parameter 'opinionCondition' was set as 'Classified as non-bullying'.

As a result of this condition, the DN2 will seek an opinion from other detection nodes only when a tweet was classified as 'non-bullying' by the DN2. When the DN2 classifies a tweet as 'bullying', it will not seek opinion from remaining detection nodes. The result of this experiments are shown in the Table 4-8.

Table 4-8: Performance of DN2 in Case 5 with respect to no-collaboration case and Case 2 – Three node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.40	25	0.56	0.60
Case 2	65.81	45.12	96.61	2.10
Case 5	65.81	45.12	96.61	1.98

Collecting opinions only when the tweet is classified as non-bullying has positive impact on time taken for classification. Additionally, it does not affect precision and recall achieved in Case 2. In the non-collaboration mode the DN2 is taking 0.60 seconds for classifying 626 tweets. That is, it is taking an average of 0.95 milliseconds to classify a tweet. When taking an opinion every time, approximately each tweet takes 3.35 milliseconds for classification. The DN2 only classifies 4 tweets as 'bullying'; for all the other tweets, it is taking an opinion from others for classification. The expected time taken to classify all tweets in Case 5 when collaboration only happens when a tweet is classified as non-bullying is 2.08 seconds. The algorithm took 1.98 seconds, which is close to the expected value.

Case 6:

Cases 1 through 5 show performance improvements in terms of recall and precision for the DN2. As it can be seen in all the above cases, the classification time has increased in each case compared to the no collaboration case. The intention of this case was to reduce to classification time while achieving better recall and precision. Variable parameters used in Case 6 are indicated in Table 4-3. The variable parameter 'opinionCondition' was set as 'Classified as non-bullying'.

Table 4-9: Performance of DN2 in Case 6 with respect to no-collaboration case and Case 1 – Three node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.40	25	0.56	0.60
Case 1	64.54	44.22	97.29	2.12
Case 6	64.60	44.25	96.95	1.92

It can be seen from results shown in Table 4-9, recall and precision were improved for the DN2 similar to Case 1 while reducing the time (7% compared to Case 1). Similar to Case 5, collecting opinions only when a tweet is classified as non-bullying has positive impact on time taken for the classification.

Case 7:

The variable parameter 'numberOfOpinions' was set as '2' in Case 7, while keeping all other parameters identical as in Case 6. This case can be viewed as a time reduction technique for Case 3.

Table 4-10: Performance of DN2 in Case 7 with respect to no-collaboration case and Case 3 – Three node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.40	25	0.56	0.60
Case 3	63.73	43.69	97.74	2.00
Case 7	63.73	43.69	97.74	1.93

Similar to Cases 5 and 6, there is a limited improvement in classification time as seen in Table 4-10.

Results of Case 1 to Case 7 experiments are consolidated in table 4-11.

Table 4-11: Results of collaboration Case 1 to Case 7 with DN2 acting as entry point – Three node network

Average	Opinions	Condition	Merging	Accuracy (%)	Precision (%)	Recall (%)	Time (ms)
Ref.	NA	NA	NA	71.40	25	0.56	0.60
Case 1	One	Always	OR	64.54	44.22	97.29	2.12
Case 2	Two	Always	AND	65.81	45.12	96.61	2.10
Case 3	Two	Always	OR	63.74	43.69	97.74	2.00
Case 4	Two	Always	Majority	65.97	45.24	96.61	2.10
Case 5	Two	Classified as non-bullying	AND	65.81	45.12	96.61	1.98
Case 6	One	Classified as non-bullying	OR	64.60	44.25	96.94	1.92
Case 7	Two	Classified as non-bullying	OR	63.73	43.69	97.74	1.93

It can be observed from the above table that both recall and precision of the bad performing node, DN2 has been increased in all the collaboration cases. Additionally, collaboration has increased classification time. Collaboration has proved to be effective in this set of experiments with a single bad performing node. It can be clearly seen from the graph in Figure 4-3 that collaboration configurations do help in increasing the performance of the DN2.

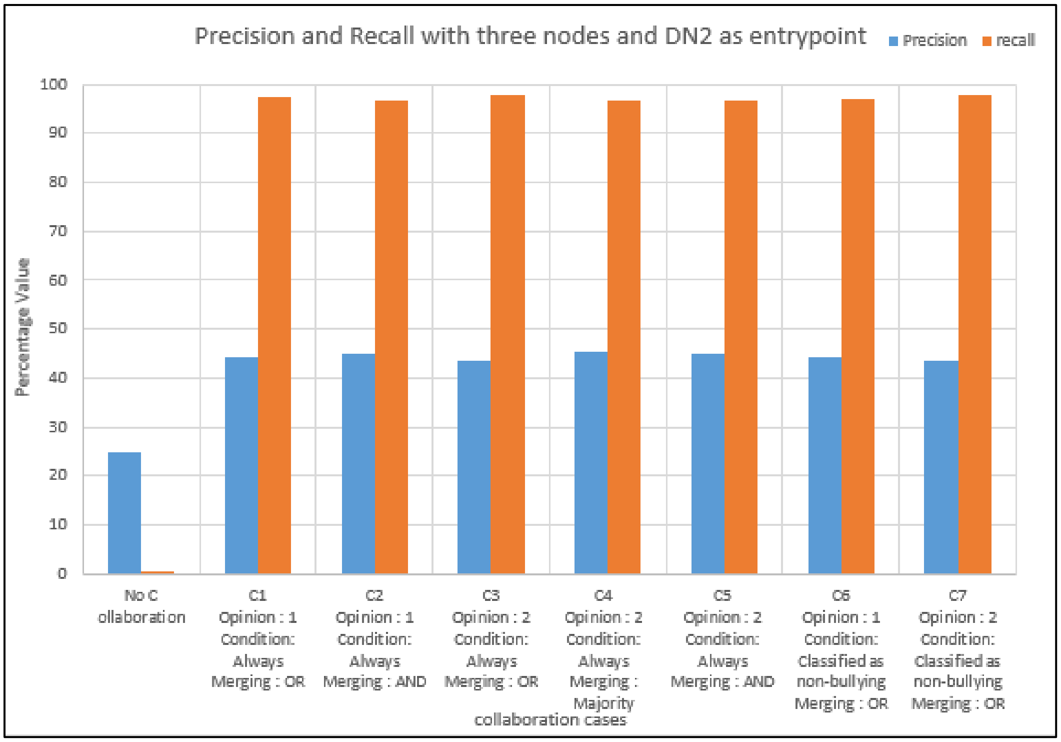


Figure 4-3: Precision and recall for collaboration cases with three nodes in the network and DN2 acting as entry point – Three node network

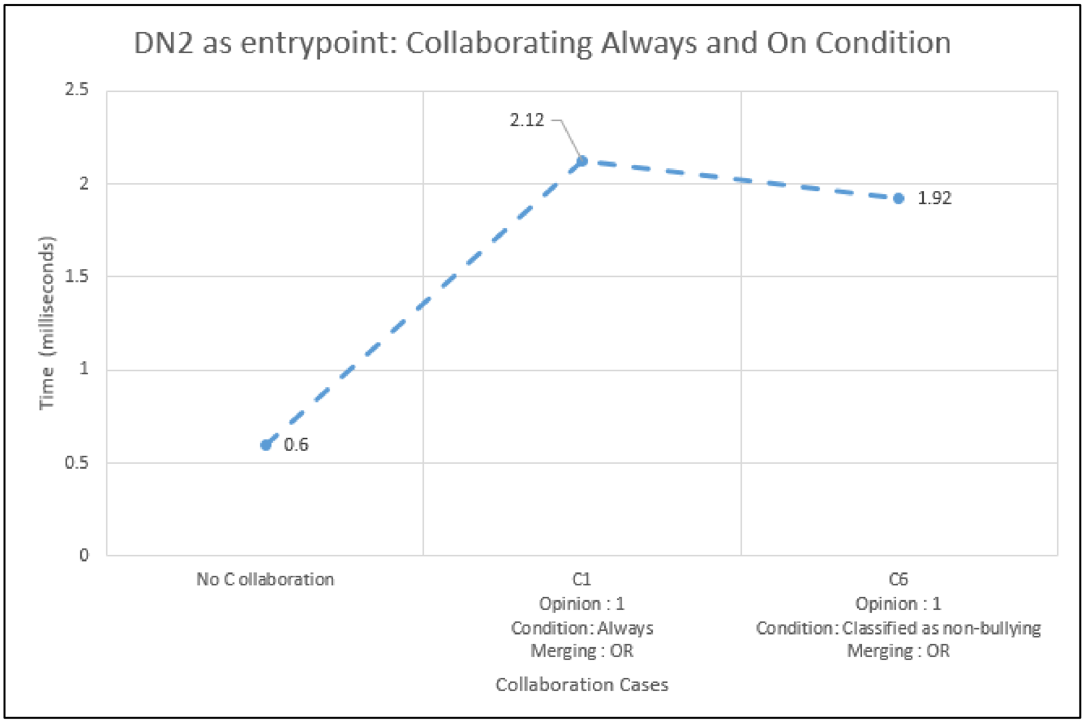


Figure 4-4: Classification time of ‘always’ collaborating detection node and collaborating only when classified as non-bullying – Three node network

The Figure 4-4 shows that, as expected, the collaboration approach has increased the total classification time. The classification time recorded for Case 1, in which opinions were sought for every tweet, is 7% higher than the classification time recorded for Case 6 in which opinions were sought when a tweet was classified as ‘non-bullying’. The graph shows that the approach of taking opinions only when the tweet is classified as ‘non-bullying’ is only marginally reducing classification time for a bad performing node.

An identical set of experiments was repeated with DN3 acting as entry point, which is a better performing node in this network. The DN3 took opinions from DN1 and DN2. There was only one node (DN2) in this network that was having poor recall and precision. It was expected that a single bad performing node’s opinion should not significantly bring down performance of a better performing node. The parameter table was updated as shown in Table 4-12. The results of this configuration are compiled in a Table 4-13.

Table 4-12: Fixed parameter values for Case 1 to Case 7 when DN3 is acting as entry point – Three node network

Parameter Name	Parameter Value
host	DN3
isEntryPoint	True for DN3 and False for others
Model	D/3 that is performing better on current test set
Algorithm	Logistics
detectionNodeList	DN1 and DN2
selectionType	Random

Table 4-13: Performance of entry point DN3 for collaboration cases 1 to 7 – Three node network

Averge	Opinions	Condition	Merging	Accuracy	Precision	Recall	Time
Reference	NA	NA	NA	65.33	44.76	96.61	0.56
Case 1	One	Always	OR	64.50	44.21	97.51	2.06
Case 2	Two	Always	AND	65.33	44.79	97.17	2.01
Case 3	Two	Always	OR	63.73	43.69	97.74	2.13
Case 4	Two	Always	Majority	65.97	45.24	96.61	2.05
Case 5	Two	Classified as non-bullying	AND	65.33	44.79	97.17	1.30
Case 6	One	Classified as non-bullying	OR	64.50	44.21	97.40	1.22
Case 7	Two	Classified as non-bullying	OR	63.73	43.69	97.74	1.18

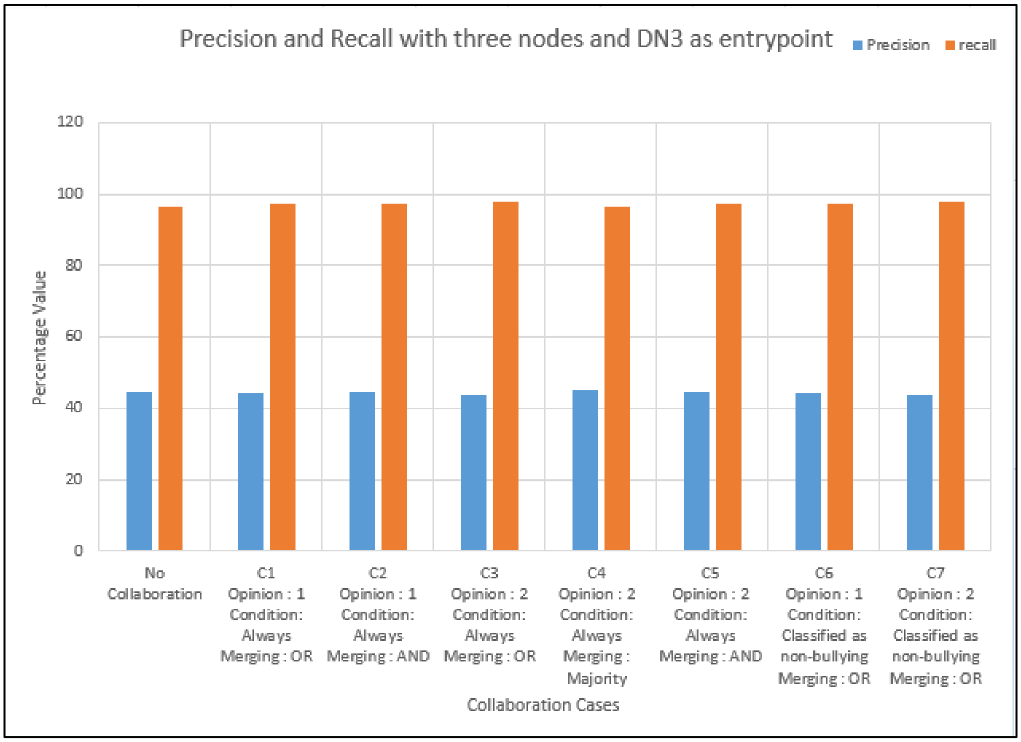


Figure 4-5: Precision and recall for collaboration cases with three nodes in the network and DN3 acting as entry point– Three node network

Initiating collaborations with a better performing node when a bad performing node exists in the network, has slightly (1%) affected precision in Case 3. The decrease in the precision is due to the fact that in 'OR' merging, the false positives are the union of false positives from all the detection nodes. For other cases it was observed that precision and recall were not decreased.

The entry point was a better performing node, thus, at least one better performing node's opinion is always contributing to the merged results. This is the reason that the 'OR' merging technique results were close to be a better performing node's results as in Case 1, Case 3, Case 6, and Case 7.

As there was only one bad performing node present in this network. The 'Majority' merging results were also close to a better performing node's results. In the case of 'AND' Merging, without collaboration the DN3 has recall of 96.6%. It asks opinions 244 times when it classifies a tweet as non-bullying, thus recall does not drop in this operation. Amongst the opinions asked, only a single tweet is corrected by the collaboration. This improves recall from 96.6 to 97.2.

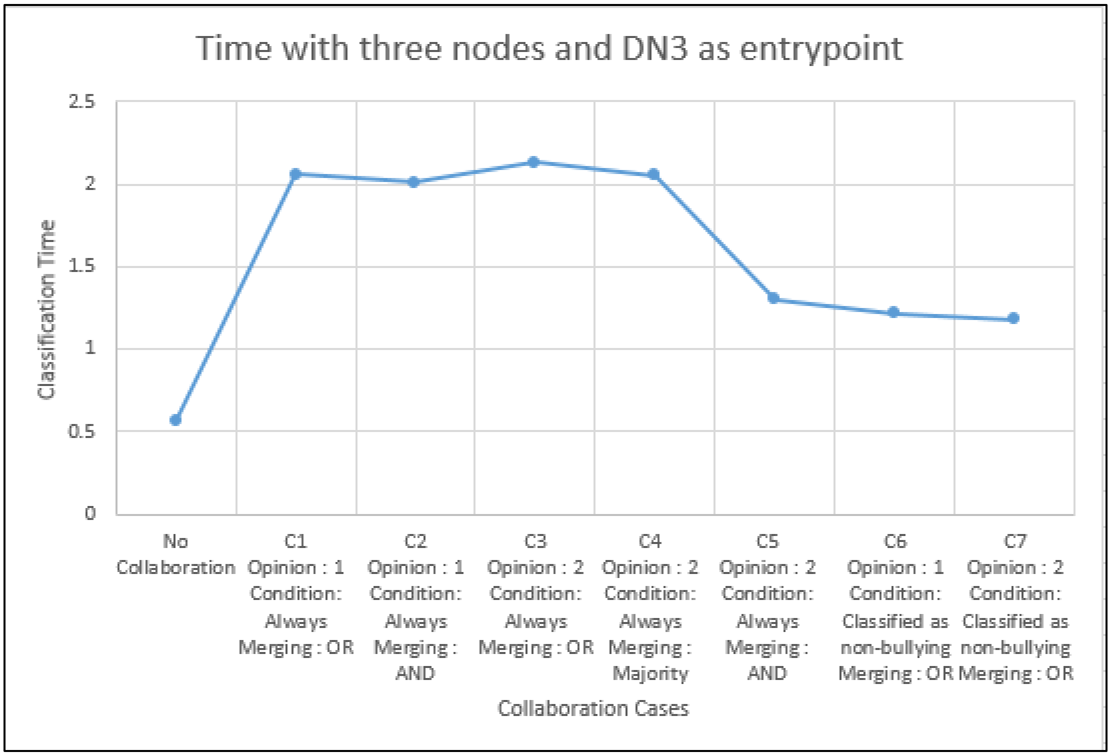


Figure 4-6: Precision and recall for collaboration cases with DN3 acting as entry point – Three node network

When not collaborating, the DN3 is taking 0.56 seconds for classifying 626 tweets. That is, it is taking an average of 0.89 milliseconds to classify a tweet. In collaboration Case 2, when an opinion is taken every time, approximately each tweet takes 3.21 milliseconds for the classification. In Case 5, the DN3 classifies 382 tweets as ‘bullying’; for all the other tweets, it is taking opinions from others for the classification. The expected time taken to classify all tweets in the Case 5 (from above two values) is 1.12 seconds; 1.30 seconds is close to this value. In short, the time reduction technique used in Case 5, Case 6, and Case 7 was effective when the entry point was a better performing node. The DN3 initiates collaboration for less number of tweets and takes less time for classification. From Figure 4-6, we can see that the Collaboration Case 6 took 40% less time than Case 1, and Case 7 took 45% less time than Case 3. In Cases 6 and 7, the detection node initiated collaboration only when the entry point classifies a tweet as ‘non-bullying’. As the DN3 has better recall, thus, it has less number of false positives.

Results were encouraging and it was concluded that for these sets of experiments, distributed-collaborative patterns used with ‘OR’, ‘Majority’, and ‘AND’ merging can be used to improve overall performance of a network even with a bad performing node. Additionally, collaborating when an entry point classifies a tweet as non-bullying appears to be a better configuration than initiating collaboration every time.

To revalidate these points we have repeated the same set of experiments for 4 DNs and 5 DNs.

4.2.2 Experimental configurations for four detection nodes

Similar to 4.2.1, for these experiments, the training set discussed in Section 3.2 was divided into disjoint four training sets by keeping the ratio of bullying over non-bullying tweets the same as the original training set. These newly created training sets were used for training four detection nodes with the logistics algorithm. To analyze the performance of the individual detection nodes, the test set was classified by each detection node individually. The classification performance of each node is compiled in Table 4-14 in terms of accuracy, precision, recall, and time. For reference, the performance of a detection node with original training set is added to the table as well.

Table 4-14: Performance of classification algorithm at individual detection nodes with split knowledge base – Four node network

Node	Accuracy (Percentage)	Precision (Percentage)	Recall (Percentage)	Time (seconds)
Detection Node 1	71.56	33.33	0.56	0.66
Detection Node 2	65.49	44.91	97.17	0.74
Detection Node 3	70.77	25	1.69	0.56
Detection Node 4	65.49	44.88	96.61	0.62
Reference	66.29	45.50	97.17	0.64

Similar to the three nodes experiments, an individual detection node will have inferior performance compared to the reference. Nodes DN1 and DN3 have less recall, hence, they are considered poor performing nodes. The DN2 and DN4 have better recall and good precision, hence, they are considered better performing nodes.

Collaboration Cases: Description, Execution, and Results

The base configuration for these experiments has four DNs discussed in the above section. Performance of collaborative configurations among these nodes was evaluated. This set of detection nodes has two poor performing nodes. As per our hypothesis, the distributed-collaborative approach must increase performance of poor performing nodes in configurations patterns and performance of better performing nodes should not be reduced significantly by collaboration. Many cases with four detection nodes in the network were created. Similar to the three nodes network, for each case, the experiment is repeated five times to validate the results. In order to normalize the results, average accuracy, precision, recall, and time were calculated. Identical parameters for Case 1 to Case 10 are listed in table below:

Table 4-15: Identical parameters used in cases from 1 to 10 in experiment – Four node network

Parameter Name	Parameter Value
Host	DN1
isEntryPoint	True for DN1 and False for all other nodes
Model	D/4 that is not performing better on current test set
Algorithm	Logistics
detectionNodeList	DN2, DN3, and DN4
selectionType	Random

The entry point for all the cases was selected as the DN1 (the least performing node). All four nodes hold one fourth of the tweets from the original training set and uses the Logistics algorithm to create classification models. The client application was adjusted to send a tweet only to the DN1 for classification. The DN1 takes opinions of DN2 and/or DN3 and/or DN4 nodes regarding classification of a given tweet. It will randomly select the node to take opinions from.

Similar to the experiments performed for three detection nodes network, collaboration parameters nuberOfOpinions, opinionCondition, and MERGE were varied to create cases 1 to 7. Details of these variable parameters were compiled in Table 4-16.

Table 4-16: Design of experiments by varying collaboration parameter values in experiments – Four node network

Cases	numberOfOpinions	opinionCondition	MERGE
Case 1 (C1)	1	True i.e. Always	OR
Case 2 (C2)	2	Always	OR
Case 3 (C3)	3	Always	OR
Case 4 (C4)	2	Always	Majority
Case 5 (C5)	3	Always	Majority
Case 6 (C6)	2	Classified as non-bullying	OR
Case 7 (C7)	2	Classified as non-bullying	Majority
Case 8 (C8)	2	Always	AND
Case 9 (C9)	3	Always	AND
Case 10 (C9)	2	Classified as non-bullying	AND

Case 1:

In this case, the DN1 will randomly seek an opinion from one of other three DNs every time. 'OR' merging technique is used. The results of this experiment are compiled in the Table 4-17:

Table 4-17: Performance of DN1 in Case 1 with respect to no-collaboration case – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 1	67.86	45.45	67.68	2.35

The recall of the DN1 is increased due to collaboration. However, it was not close to better performing nodes. When DN1 selected to take opinion from better performing nodes, they are overriding the wrong classification by the DN1. However, the DN1 takes opinions from a bad performing node, so it may provide an incorrect result. Overall recall for the DN1 has increased significantly due to opinions from better performing nodes when compared to no-collaboration. Precision is increased as a result of a higher number of true positives, and opinions are taken from nodes that have precision greater than 25%. Additionally, collaboration has increased the classification time.

Case 2:

For the Case 2, the only parameter that is changed from the Case 1 is numberOfOpinions. The DN1 takes two opinions before classifying a tweet every time. The results of this experiment are compiled in the Table 4-18:

Table 4-18: Performance of DN1 in Case 2 with respect to no-collaboration case – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 2	64.31	44.06	97.29	2.11

The recall of the DN1 is increased with this collaboration case and it is close to one of the better performing nodes. This is due to the fact that the DN1 is taking two opinions in a network. When the DN1 chooses two out of these three nodes, at least one opinion is coming from a detection node with better recall. As we have used ‘OR’ merging technique, even if one of the nodes classifies a tweet as ‘bullying’, it is considered as ‘bullying’. As most of the time this classification by a better recall node will be correct, recall of the DN1 is increased. However, ‘OR’ opinion merging combines false positives from detection nodes as discussed in the three detection node study. This causes precision to slightly drop from Case 1. Finally, the effect on the classification time is not drastically increased due to the fact that an extra opinion is taken, as the opinions are asked in parallel.

Case 3:

In Case 3, the numberOfOpinions is again increased and made three. For every tweet, the DN1 is now taking opinion from all the nodes in the network. The opinions are merged using ‘OR’ merging. The results of this experiment are compiled in the Table 4-19:

Table 4-19: Performance of DN1 in Case 3 with respect to no-collaboration case – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 3	63.90	43.80	97.74	2.16

The case has similar results as Case 2. The recall of the DN1 is increased with this collaboration case and is comparable to a better performing node. Precision is dropped slightly compared to Case 2. This is because false positives in this cases were union of false positives for all three nodes.

Case 4:

In Case 4, the numberOfOpinions is set to two and the 'Majority' merging technique is used. For every tweet, the DN1 is now taking opinions from two out of three other detection nodes in the network. If at least 2 opinions (including its own) classifies a tweet as 'bullying' then only the DN1 classifies that tweet as 'bullying'. The results of this experiment are compiled in the Table 4-20:

Table 4-20: Performance of DN1 with respect to no-collaboration case – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 4	69.62	44.94	33.11	2.25

The recall of the DN1 is increased with this collaboration case; however, it is not comparable to better performing nodes. This is due to the fact that there are two nodes with better recall and two nodes with poor recall. Nodes to take opinion from are selected at random by the DN1. Hence, every time there is at least one node with poor recall in merged results. Due to the 'Majority' merging, correctly classifying a tweet as 'bullying' happens only when both the nodes chosen by the DN1 are better performing. 'Majority' merging has increased precision by reducing number of false positives.

Case 5:

For Case 5, the numberOfOpinions is set to two and the ‘Majority’ merging technique is used. For every tweet, the DN1 is now taking opinion from all three detection nodes in the network. If more than 2 opinions (more than half) are classifying a tweet as ‘bullying,’ then only the DN1 classifies that tweet as ‘bullying’. The results of this experiment are compiled in the Table 4-21:

Table 4-21: Performance of DN1 in Case 5 with respect to no-collaboration case – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 5	72.04	75	1.69	2.19

The recall of the DN1 is increased by less than 1% in this collaboration case; the case fails to improve the recall of the DN1. When opinions of all detection nodes have been taken into consideration, even if two nodes with better recall correctly classify a tweet as ‘bullying’, as they do not form a ‘Majority’, the resultant classification is overridden by poor recall nodes.

We repeated this case by changing the ‘Majority’ condition. The new condition is at least 2 (at least half of) opinions should confirm that a tweet is ‘bullying’ for DN1 to classify the tweet as ‘bullying’. The results of this experiment are compiled in the Table 4-22:

Table 4-22: Performance of DN1 in Case 5 with slightly modified Majority condition with respect to no-collaboration case – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 5	65.65	45	96.61	2.51

It can be seen from the results that this change in the Majority condition yields better recall and precision for Case 5. The condition will allow the distributed-collaborative approach to work even if half of the nodes are bad performing in the network.

Case 6:

Case 6 is similar to Case 2, except for the collaboration parameter ‘opinionCondition’. The DN1 initiated collaboration only when it classified a tweet as ‘non-bullying’. The parameter was modified to see its effect on the classification time. The results of this experiment are compiled in the Table 4-23:

Table 4-23: Performance of DN1 in Case 6 with respect to no-collaboration case and case 2 – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 2	64.31	44.06	97.29	2.11
Case 6	64.76	44.40	97.63	2.14

Performance results are similar to the Case 2; however, there is no time improvement seen in this case, due to the fact that the DN1 has poor recall. Hence, it is classifying only four tweets as ‘bullying’; for all the other tweets, it is taking opinions from other nodes for classification. Therefore, a significant improvement is not observed in the classification time.

Case 7:

Case 7 is similar to Case 4, except DN1 initiates collaboration only when it classifies a tweet as ‘non-bullying’. This parameter is modified by aiming to reduce time required by DN1. The results of this experiment are compiled in Table 4-24:

Table 4-24: Performance of DN1 in Case 7 with respect to no-collaboration case and case 4 – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 4	69.62	44.94	33.11	2.25
Case 7	69.20	44.92	34.30	2.23

Performance results are similar to Case 4, but the significant time improvement is not seen for the same reason explained in Case 6.

Case 8:

In this case, ‘numberOfOpinions’ is set to two and ‘AND’ merging is used. It is expected that with a bad performing node in the network ‘AND’ merging will have reduced recall.

Table 4-25: Performance of DN1 in Case 8 with respect to no-collaboration case – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 8	69.33	44.23	32.54	2.52

Although the precision and recall is improved over non-collaboration case, it is not closer to that of any better performing node. ‘AND’ merging is not only reducing false positives in this case, it is also adding more numbers of false negatives by overriding correct classification from individual nodes.

Case 9:

In this case, ‘numberOfOpinions’ is set to three and all the other parameters are similar to Case 8. This case analyzes the effect of merging all the opinions with ‘AND’ merging when a bad node is present in the network.

Table 4-26: Performance of DN1 in Case 9 with respect to no-collaboration case – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 9	71.73	50.00	1.69	2.43

Recall is improved only one percent over the non-collaboration case. This was expected, as a single bad performing node in the network overrides the correct assessments by other nodes with ‘AND’ merging. Thus, it can be observed here that using ‘AND’ merging with a bad performing node in the network will not yield better recall if the system collects all the opinions.

Case 10:

In this case, all the other parameters are similar to Case 8 except the opinion condition. This case assesses the impact on the classification time when the DN1 will seek opinions when it classifies a tweet as ‘non-bullying.

Table 4-27: Performance of DN1 in Case 10 with respect to no-collaboration case – Four node network

Average for	Accuracy	Precision	Recall	Time
No Collaboration	71.56	33.33	0.56	0.66
Case 10	69.94	46.05	35.93	2.39

Precision and Recall are similar to Case 8. However, significant improvement is not noticed in the classification time for the same reasons explained in Case 6.

This concludes experiments for the collaboration cases where the DN1 is acting as entry point and taking opinions from other detection node. Consolidated results are listed in Table 4-28.

Table 4-28: Results of collaboration Case 1 to Case 10 with DN1 acting as entry point – Four node network

Average	Opinions	Condition	Merging	Accuracy (%)	Precision (%)	Recall (%)	Time (ms)
Reference	NA	NA	NA	71.56	33.33	0.56	0.66
Case 1	One	Always	OR	67.86	45.45	67.68	2.35
Case 2	Two	Always	OR	64.31	44.06	97.28	2.11
Case 3	Three	Always	OR	63.90	43.80	97.74	2.16
Case 4	Two	Always	Majority	69.61	44.94	33.11	2.25
Case 5	Three	Always	Majority	72.04	75.00	1.69	2.19
Case 5 (Subcase)	Three	Always	Majority	65.65	45	96.61	2.51
Case 6	Two	Classified as non-bullying	OR	64.76	44.40	97.62	2.14
Case 7	Two	Classified as non-bullying	Majority	69.52	44.92	34.69	2.23
Case 8	Two	Always	AND	69.33	44.23	32.54	2.52
Case 9	Three	Always	AND	71.73	50.00	1.69	2.43
Case 10	Two	Classified as non-bullying	AND	69.94	46.05	35.93	2.39

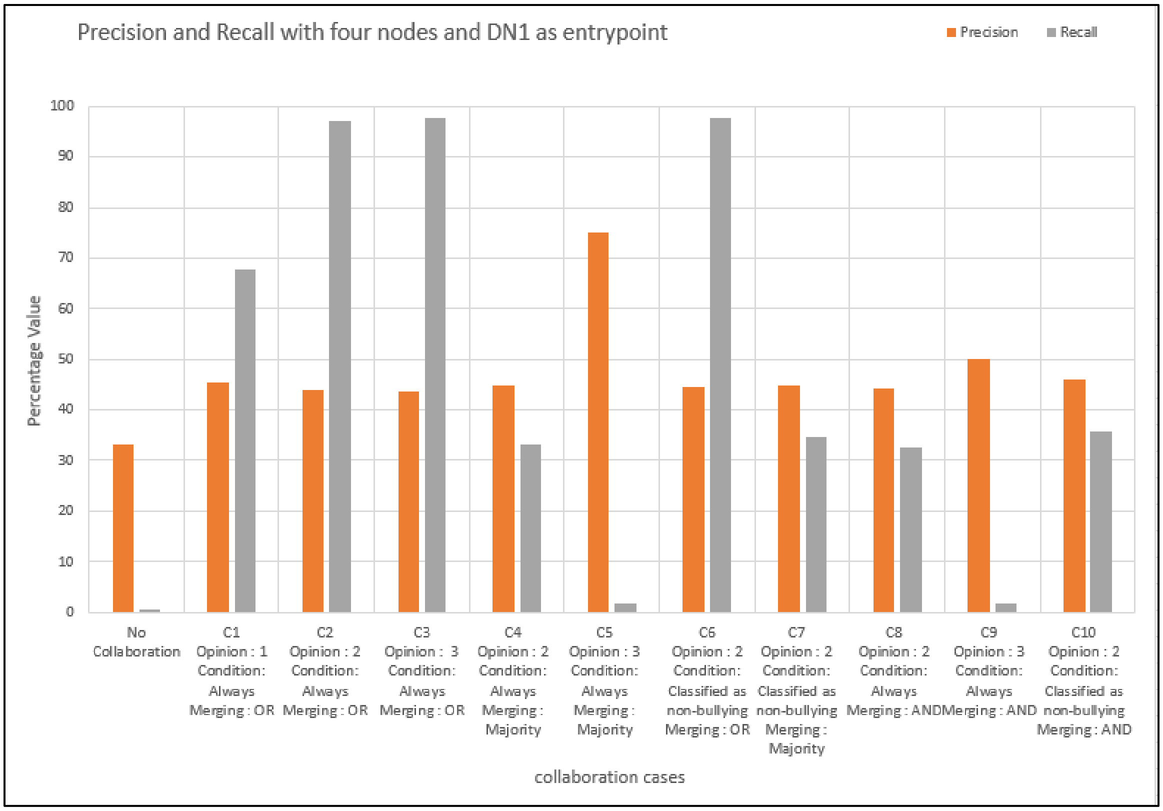


Figure 4-7: Precision and recall for collaboration cases DN1 acting as entry point – Four node network

Compared to the no-collaboration case, all the collaboration patterns significantly increase the recall and precision for the DN1, except for Cases 5 and 9. This indicates the fact that the 'Majority' and 'AND' techniques are more affected by the presence of bad performing nodes. 'Majority' merging will not work when half of the detection nodes in the network are poor performing. 'AND' merging will not work even if a single detection node in the network is a bad performing node.

In cases of 'OR' merging, the performance of the bad performing entry point was progressively improved as the number of opinions were increased from one to three. When the number of opinions were more such as three, the adverse effect of the poor performing detection node was mitigated. However, every additional opinion taken has decreased the precision due to the addition of more false positives. Although, this decrease in precision is not significant in this case, the optimum number of opinions needs to be collected to balance the precision and recall.

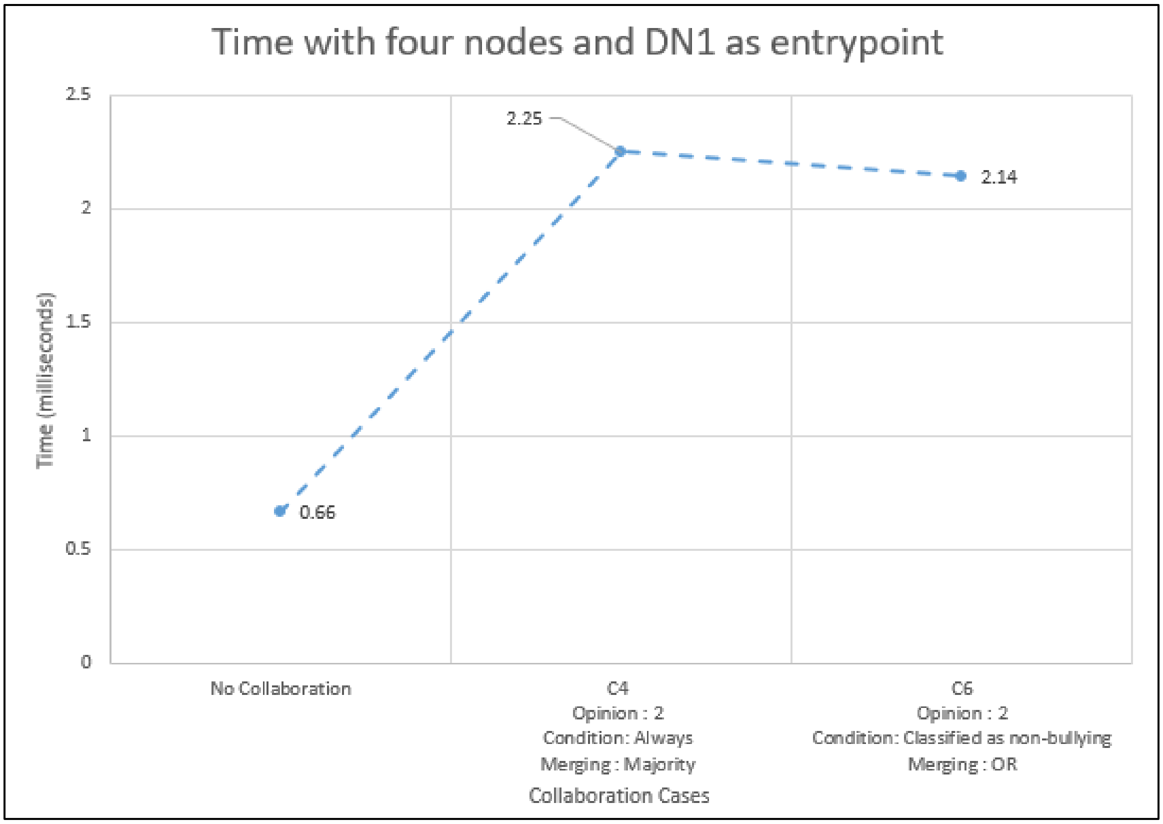


Figure 4-8: Classification time of ‘always’ collaborating detection node and collaborating only when classified as non-bullying– Four node network

Similar to networks with three nodes, in this situation collaboration increases the classification time. The time reduction technique used in Cases 6 and 7 was not effective when the entry point is a poor performing node. This confirms findings from the three node network.

An identical set of experiments was repeated with the DN4 acting as the entry point, which is a better performing node in this network. The DN4 took opinions from DN1, DN2, and DN3. The intention of these experiments was to study the effect of this network configuration on a better performing node seeking opinions from other nodes. The fixed parameter table was modified as shown in Table 4-29. The results of this configuration are compiled in Table 4-30 below.

Table 4-29: Fixed parameter values when DN4 is acting as entry point – Four node network

Parameter Name	Parameter Value
host	DN4
isEntryPoint	True for DN4 and False for others
Model	D/4 that is performing better on current test set
Algorithm	Logistics
detectionNodeList	DN1, DN2, and DN3
selectionType	Random

Table 4-30: Collaboration cases performance when DN4 is acting as entry point – Four node network

Averge	Opinions	Condition	Merging	Accuracy	Precision	Recall	Time
Ref.	NA	NA	NA	65.49	44.88	96.61	0.623
Case 1	One	Always	OR	64.98	44.55	97.40	2.18
Case 2	Two	Always	OR	64.38	44.13	97.63	2.64
Case 3	Three	Always	OR	63.90	43.80	97.74	2.37
Case 4	Two	Always	Majority	66.64	43.96	65.54	2.31
Case 5	Three	Always	Majority	72.04	75	1.69	2.19
Case 6	Two	Classified as non-bullying	OR	64.50	44.22	97.63	1.40
Case 7	Two	Classified as non-bullying	Majority	65.65	45.03	97.17	1.43
Case 8	Two	Always	AND	69.33	44.23	32.54	2.52
Case 9	Three	Always	AND	71.73	50.00	1.69	2.53
Case 10	Two	Classified as non-bullying	AND	65.65	45.03	97.18	1.46

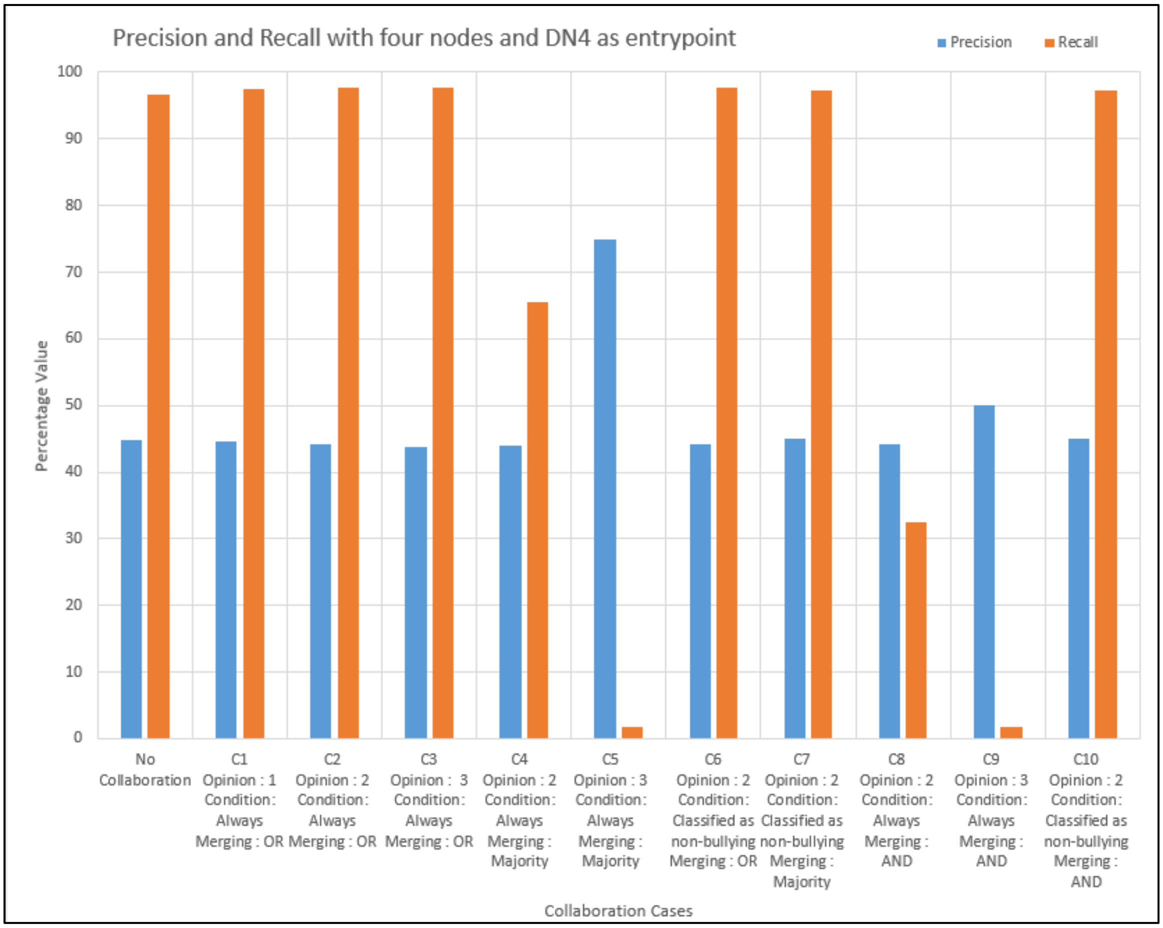


Figure 4-9: Precision and recall when DN4 is acting as entry point for collaboration Case 1 to Case 10 – Four Node Network

Similar to the three node network, recall of the better performing node has not decreased in cases of ‘OR’ merging, even if a bad performing node exists in the network. With any number of opinions taken, at least one better performing node’s opinion is always contributing to the merged results. Precision was affected less than one percent due to the addition of false positive as we increase the number of opinions.

There were two bad performing nodes present in this network. In the case of ‘Majority’ the merging in Case 4, a good performing node is collecting opinions from the network that has 2 bad performing nodes and a single good performing node. This reduced the chances of correct classification forming a majority. Thus, recall with collaboration (Case 4) was dropped to 65%. When 3 opinions were taken, two out of four opinions were always coming from bad performing nodes. Hence, correct classifications by better performing nodes were never in the ‘Majority’. Hence, we do not recommend ‘Majority’ merging when at least half of the nodes are bad performing nodes.

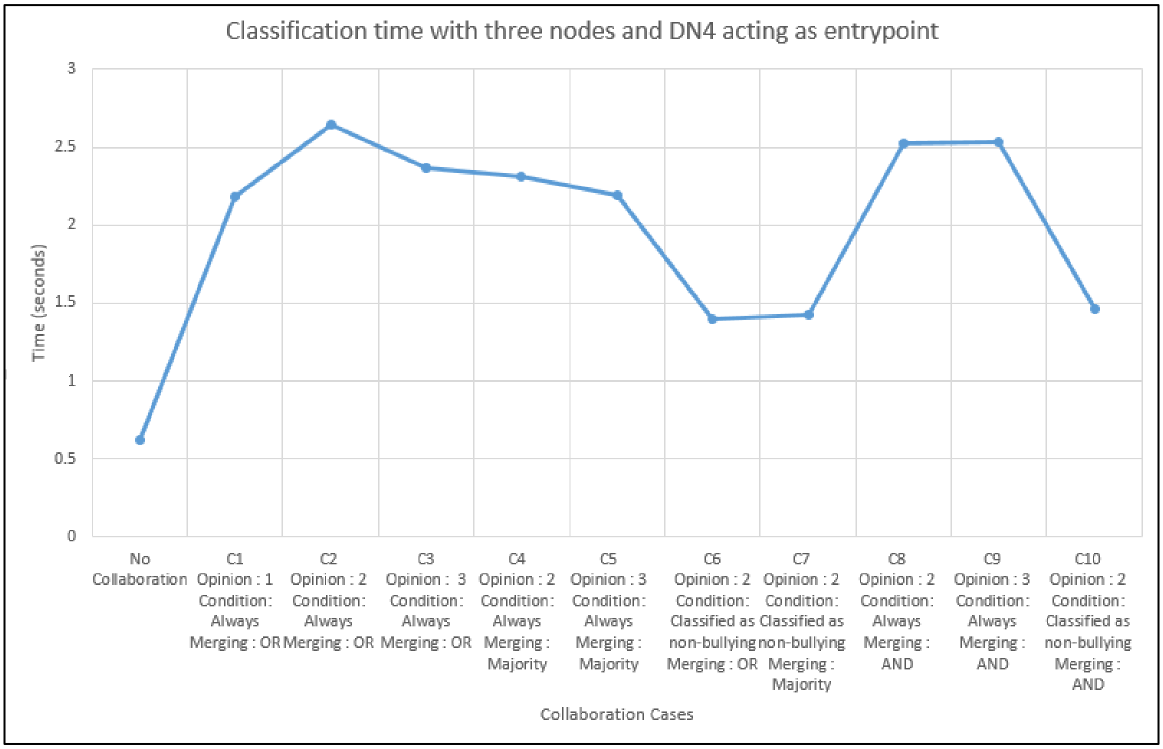


Figure 4-10: Classification time when DN4 is acting as entry point for collaboration Case 1 to Case 10 – Four Node Network

Similar to networks with three nodes, the time reduction technique used in Cases 6 and 7 was effective when entry point is a better performing node. Classification time recorded for Case 6 is 47% lesser than Case 2. This confirms our finding from the three node network.

4.2.3 Experimental configurations for five detection nodes

Similar to experiments performed in Section 4.2.1, for these experiments, the training set discussed in Section 3.2 was divided into five disjoint training sets by keeping the ratio of bullying over non-bullying tweets the same as the original training set. These newly created training sets were used for training five detection nodes with the logistics algorithm. To analyze the performance of individual detection nodes, a test set was classified by each detection node individually. The classification performance of each node is compiled in

Table 4-31 in terms of accuracy, precision, recall, and time. For reference, the performance of a detection node with original training set is added in the table.

Table 4-31: Performance of individual detection nodes in the network – Five node network

Node	Accuracy (Percentage)	Precision (Percentage)	Recall (Percentage)	Time (seconds)
Detection Node 1	65.34	44.76	96.61	0.66
Detection Node 2	71.73	50.00	0.56	0.59
Detection Node 3	64.38	44.10	97.18	0.63
Detection Node 4	71.57	0.00	0.00	0.62
Detection Node 5	71.41	33.33	1.13	0.64
Reference	66.29	45.50	97.17	0.64

Similar to three and four nodes experiments, individual detection nodes will have inferior performance compared to the reference. The DN2, DN4 and DN5 have much less recall, hence, they are considered to be poor performing nodes. The DN1 and DN3 have better recall and good precision, hence, they are considered to be better performing nodes.

Collaboration Cases: Description, Execution, and Results

The base configuration for these experiments has five DNs discussed in the above section. Performance of collaborative configurations among these nodes was evaluated. This set of detection nodes has three poor performing nodes. As per our hypothesis, the distributed-collaborative approach must increase performance of poor performing nodes in collaborative patterns, and the performance of better performing nodes should not be reduced significantly by collaboration. Different cases with five detection nodes in the network were created. Similar to three and four node networks, for each case, the experiment is repeated five times to validate the results. In order to normalize the results, average accuracy, precision, recall, and time were calculated. Identical parameters for Case 1 to 10 are listed in the Table 4-32.

Table 4-32: Fixed parameter values when DN4 is acting as entry point - Five nodes network

Parameter Name	Parameter Value
Host	DN4
isEntryPoint	True for DN4 and False for all other nodes
Model	D/5 that is not performing better on current test set
Algorithm	Logistics
detectionNodeList	DN1, DN2, DN3 and DN5
selectionType	Random
OpinionCondition	Classified as non-bullying

The entry point for all the cases was selected as the DN4 which has the least recall. All five nodes use one fifth of the tweets from the original training set, and use the Logistics algorithm to create the classification model. The client application has been adjusted to send a tweet only to the DN4 for classification; DN4 then takes the opinions from the DN1 and/or the DN2 and/or the DN3 and/or the DN5 regarding classification of the given tweet. It will randomly select the node to take opinion from. From the observations made in three nodes network and four nodes network, it can be concluded that, collecting opinions when entry point classifies a tweet as ‘non-bullying’ appears to impact time positively without reducing precision and recall. Hence, for the five node network opinions were taken only when the entry point classifies a tweet as ‘non-bullying’.

Similar to experiments performed for three and four detection node networks, collaboration parameters numberOfOpinions and MERGE were varied to create Cases 1 to 10. Details of these variable parameters were compiled in Table 4-33. Results of these experiments are compiled in Table 4-34.

Table 4-33: Design of experiments by varying collaboration parameter values – Five node network

Cases	numberOfOpinions	MERGE
Case 1 (C1)	1	OR
Case 2 (C2)	2	OR
Case 3 (C3)	3	OR
Case 4 (C4)	4	OR
Case 5 (C5)	2	AND
Case 6 (C6)	3	AND
Case 7 (C7)	4	AND
Case 8 (C8)	2	Majority
Case 9 (C9)	3	Majority
Case 10 (C9)	4	Majority

Table 4-34: Collaboration cases performance when DN4 is acting as entry point – Five node network

Average	Opinions	Merging	Accuracy (%)	Precision (%)	Recall (%)	Time (s)
Reference	NA	NA	71.57	0.00	0.00	0.62
Case 1	1	OR	66.61	42.01	47.57	3.24
Case 2	2	OR	65.02	43.65	81.47	2.87
Case 3	3	OR	63.71	43.65	97.51	2.97
Case 4	4	OR	63.26	43.36	97.74	2.86
Case 5	2	AND	70.77	45.78	18.42	3.01
Case 6	3	AND	71.57	0.00	0.00	2.96
Case 7	4	AND	71.57	0.00	0.00	2.92
Case 8	2	Majority	70.58	44.81	17.74	3.10
Case 9	3	Majority	68.12	44.37	50.17	3.04
Case 10	4	Majority	71.73	50.00	0.56	2.93

In Cases 1 to 4, the ‘OR’ merging technique was used while increasing the number of opinions from 1 to 4. This network contains two good performing and two bad performing nodes providing the opinions to the entry point detection node. It can be observed from the results that as the number of opinions collected increased, the recall was increased. Increase in the recall was prominent in Cases 1 through 3, while the little increase was observed in the Case 4. As the number of opinions increases from 1 to 4, the probability of good performing nodes providing the input increases progressively, consequently, increasing the recall of the distributed-collaborative network. It was also observed from the results that the precision was significantly improved as a results of distributed-collaboration when compared to the no collaboration approach. However, the number of opinions has little to no effect (<1%) on the precision in the case of the OR merging technique.

In the case of the AND merging technique (from Cases 5 to 7), it was observed that the recall and the precision increased when compared with the no collaboration approach with 2 opinions. However, both the precision and the recall decrease to 0 as the number of opinion increases. As the number of opinion increases, the probability of a bad performing node contributing to the results increases, thereby, decreasing the recall and the precision in later cases.

In the case of the Majority merging technique (Cases 8 to 10), it was observed that precision increased when compared to no collaboration approach and it is comparatively constant with a varying number of opinions. This observation can be justified by the reduction in the false positives due to the Majority merging technique. Recall was increased from 2 opinions to 3 opinions and then reduced to 0 for 4 opinions. The probability of correct assessments forming a majority increases as the number of opinions increases from 2 to 3. However, with 4 opinions, good performing nodes are in minority in the network. As a result, the correct assessments are being overridden by incorrect assessments.

An identical set of experiments was repeated with the DN3 acting as the entry point which is a better performing node in this network. The DN3 took opinions from the DN1, DN2, DN4 and DN5. The intention of these experiments was to study the effect this network configuration on a better performing node seeking opinions from other nodes. The fixed parameter table was modified as shown in Table 4-35. Results from this configuration are compiled in table 4-36.

Table 4-35: Fixed parameter values when DN3 is acting as entry point - Five nodes network

Parameter Name	Parameter Value
host	DN3
isEntryPoint	True for DN3 and False for all other nodes
Model	D/5 that is not performing better on current test set
Algorithm	Logistics
detectionNodeList	DN1, DN2, DN3 and DN5
selectionType	Random
OpinionCondition	Classified as non-bullying

Table 4-36: Collaboration cases performance when DN3 is acting as entry point – Five node network

Average	Opinions	Merging	Accuracy (%)	Precision (%)	Recall (%)	Time (s)
Reference	NA	NA	64.38	44.10	97.18	0.63
Case 1	1	OR	63.99	43.83	97.18	1.58
Case 2	2	OR	63.61	43.58	97.40	1.57
Case 3	3	OR	63.35	43.41	97.51	1.57
Case 4	4	OR	63.26	43.36	97.74	1.52
Case 5	2	AND	63.71	43.66	97.51	1.56
Case 6	3	AND	64.38	44.10	97.18	1.70
Case 7	4	AND	64.38	44.10	97.18	1.62
Case 8	2	Majority	64.28	44.03	97.18	1.63
Case 9	3	Majority	64.15	43.94	97.18	1.60
Case 10	4	Majority	64.38	44.10	97.18	1.58

From above results, it can be concluded that the precision and recall were not adversely affected due to the distributed-collaborative approach.

4.3 Distributed-Collaborative Detection to Using Multiple Algorithms

Experiments performed in the previous section used a common algorithm in all the DNs. That is, the same machine learning algorithm and logistics was used for the model creation by all the nodes in the network. Hence, it can be termed as a homogeneous network. In literature, however, other machine learning algorithms such as Naive Bayes and Support Vector Machine (SVM) have been successfully used by researchers for text classification. These algorithms may have better performance on different parts of the diverse test set such as Twitter. Instead of depending on a single algorithm, the detection node network can use a collection of nodes, each with a different machine learning algorithm for the model creation. This section describes the use of distributed-collaborative approach in such a heterogeneous network.

4.3.1 Experimental configurations for five detection node with heterogeneous network

For these experiments, the training set discussed in Section 3.2 was divided into five training sets by keeping the ratio of bullying over non-bullying tweets the same as the original training set. Three models were created using the logistics algorithm. Two models were created using the Naive Bayes algorithm. Additionally, models used for the DN3 and the DN5 use the same tweets, and they only use different machine learning algorithm for the model creation.

Individual performances of these models on a given test set is shown in the table below.

Table 4-37: Performance of individual detection nodes in the network – Five node heterogeneous network

Node	Accuracy (Percentage)	Precision (Percentage)	Recall (Percentage)	Time (seconds)
Detection Node 1	61.50	42.12	96.61	0.77
Detection Node 2	71.73	50.00	0.56	0.59
Detection Node 3	62.14	42.54	96.61	0.74
Detection Node 4	71.57	0.00	0.00	0.62
Detection Node 5	71.41	33.33	1.13	0.64

The experiment conducted had five detection nodes similar to the experiment performed in Section 4.2.3. Fixed parameters for each of these detection node are shown table below.

Table 4-38: Fixed parameters for detection nodes in heterogeneous network – Five nodes heterogeneous network

Parameter Name	DN 1	DN 2	DN 3	DN 4	DN 5
host	DN 1	DN 2	DN3	DN4	DN 5
isEntryPoint	False	False	False	True	False
Model	D/5	D/5	D/5	D/5	D/5
Algorithm	Naive Bayes	Logistics	Naive Bayes	Logistics	Logistics
detectionNodeList	NA	NA	NA	DN1, DN2, DN3 and DN5	NA
selectionType	NA	NA	NA	Random	NA
OpinionCondition	NA	NA	NA	Classified as non-bullying	NA

The entry point for all the cases was selected as the DN4 which has the least recall. All five nodes use one fifth of the tweets from the original training set. The DN1 and DN3 use the Naive Bayes algorithm for creating classification model and the rest of the detection nodes use Logistics algorithm to create the classification model. The client application has been adjusted to send a tweet only to the DN4 for classification. The DN4 takes opinions of the DN1 and/or the DN2 and/or the DN3 and/or the DN5 nodes regarding classification of the given tweet. It randomly selects the node to take opinion from. Collaboration parameters ‘numberOfOpinions’, and ‘MERGE’ were varied to create Cases 1 to 10. The design of experiments discussed in Table 4-33 were used to perform various experiments.

Results for this configuration are compiled in Table 4-39.

Table 4-39: Collaboration cases performance when DN4 is acting as entry point – Five node network

Average	Opinions	Merging	Accuracy (%)	Precision (%)	Recall (%)	Time (s)
Reference	NA	NA	71.57	0.00	0.00	0.62
Case 1	1	OR	65.14	40.15	47.46	3.08
Case 2	2	OR	62.88	41.92	81.13	3.04
Case 3	3	OR	61.60	42.20	96.84	3.08
Case 4	4	OR	61.66	42.26	97.18	3.03
Case 5	2	AND	69.94	42.67	18.42	3.13
Case 6	3	AND	71.50	6.67	0.11	3.11
Case 7	4	AND	71.57	0.00	0.00	3.02
Case 8	2	Majority	69.84	42.27	18.08	3.22
Case 9	3	Majority	65.91	41.50	50.17	3.21
Case 10	4	Majority	71.25	28.57	1.13	3.09

By comparing Tables 4-34 and 4-39, it can be clearly seen that the results achieved are similar for both homogeneous and heterogeneous network configurations. The average time taken by the heterogeneous network cases is higher in almost all cases. This can be justified by the fact that Naive Bayes algorithm is taking more time for classification than logistics. This can be seen in Table 4-39.

An identical set of experiments was repeated with the DN3 acting as the entry point which is a better performing node in this network. The DN3 took opinion from the DN1, DN2, DN4, and DN5. Results from this configuration are compiled in Table 4-40.

Table 4-40: Collaboration cases performance for Case 1 to 10 when DN3 is acting as entry point – Five nodes heterogeneous network.

Average	Opinions	Merging	Accuracy (%)	Precision (%)	Recall (%)	Time (ms)
Reference	NA	NA	62.14	42.54	96.61	0.74
Case 1	1	OR	61.88	42.37	96.61	1.71
Case 2	2	OR	61.92	42.40	96.84	1.69
Case 3	3	OR	61.79	42.33	96.95	1.70
Case 4	4	OR	61.66	42.26	97.18	1.70
Case 5	2	AND	62.14	42.54	96.61	1.71
Case 6	3	AND	62.14	42.54	96.61	1.70
Case 7	4	AND	62.14	42.54	96.61	1.73
Case 8	2	Majority	62.14	42.54	96.61	1.72
Case 9	3	Majority	62.04	42.47	96.61	1.73
Case 10	4	Majority	62.14	42.54	96.61	1.74

Similar to the case of bad performing nodes, the results achieved are similar for both homogeneous and heterogeneous network configurations. This can be verified by looking at the result in Tables 4-36 and 4-40. Additionally, similar to previous case, the average time taken by the heterogeneous network cases was higher in almost 9 out of 10 cases. This can be justified by same reason that Naive Bayes algorithm takes more time for classification.

The results of the above experiments seem encouraging; the proposed approach is behaving as expected for both homogeneous and heterogeneous networks. However, until now the experiments performed contained three, four, and five nodes network. To validate if the proposed approach is scalable we carried out experiments with 10 nodes. The next section discusses those experiments.

4.3.2 Experimental configurations for ten detection node with heterogeneous network

For these experiments, the training set discussed in Section 3.2 was divided into ten training sets by keeping the ratio of bullying over non-bullying tweets the same as the original training set. Six classification models were created using the logistics algorithm; each of them had a dataset with a different number of tweets. Four classification models were created using the Naive Bayes algorithm; each had a dataset with a different number of tweets. In essence, the models created had overlapping information.

Individual performances of these classification models on the given test set is shown in table below:

Table 4-41: Performance of individual detection nodes in the network – Ten node heterogeneous network

Node	Accuracy (Percentage)	Precision (Percentage)	Recall (Percentage)	Time (seconds)
DN1	62.46	42.71	96.05	0.55
DN2	71.73	50.00	0.56	0.66
DN3	70.77	25.00	1.69	0.65
DN4	60.70	41.61	96.61	0.77
DN5	58.15	40.05	96.61	0.71
DN6	62.14	42.54	96.61	0.80
DN7	65.50	44.88	96.61	0.63
DN8	62.78	43.00	97.18	0.64
DN9	60.86	41.71	96.61	0.88
DN10	71.57	0.00	0.00	0.65

The entry point for all the cases was selected as the DN10 which has the least recall. The client application has been adjusted to send a tweet only to the DN10 for classification. It will randomly select a node to collect opinion from. 'OpinionCondition' was set to Classified as non-bullying, that is, every time the entry point classifies a tweet as 'non-bullying', it collects opinions from others. Collaboration parameters 'numberOfOpinions', and 'MERGE' were varied to create Cases 1 to 6. Results of these experiments are listed in the table below.

Table 4-42: Collaboration cases performance for Case 1 to 6 when DN10 is acting as entry point – Ten nodes heterogeneous network.

Average	Opinions	Merging	Accuracy (%)	Precision (%)	Recall (%)	Time (s)
Reference	NA	NA	71.57	0.00	0.00	0.62
Case 1	2	OR	61.02	41.70	95.03	5.40
Case 2	3	OR	59.84	41.10	97.06	5.08
Case 3	2	AND	65.72	41.94	55.25	5.35
Case 4	3	AND	68.82	44.43	41.36	5.08
Case 5	2	Majority	67.57	41.32	46.89	4.36
Case 6	3	Majority	63.51	43.08	90.28	5.25

Results of identical experiments performed with the DN9, a good performing node in network, acting as entry point are compiled in the table below.

Table 4-43: Collaboration cases performance for Case 1 to 6 when DN9 is acting as entry point – Ten nodes heterogeneous network.

Average	Opinions	Merging	Accuracy (%)	Precision (%)	Recall (%)	Time (s)
Reference	NA	NA	60.86	41.71	96.61	0.88
Case 1	2	OR	60.22	41.32	96.84	2.62
Case 2	3	OR	59.98	41.21	97.32	2.19
Case 3	2	AND	60.86	41.71	96.61	2.51
Case 4	3	AND	60.86	41.71	96.61	2.51
Case 5	2	Majority	60.86	41.71	96.61	2.84
Case 6	3	Majority	60.86	41.72	96.72	2.51

The results for precision, recall and time are consistent with the results obtained from previous experiments. The distributed collaborative approach is increasing precision and recall for a bad performing node without drastically hampering performance of a better performing node. The classification time results, however, raises concerns as it increases with the number of nodes in the network. Figure 4-11 shows this co-relation.

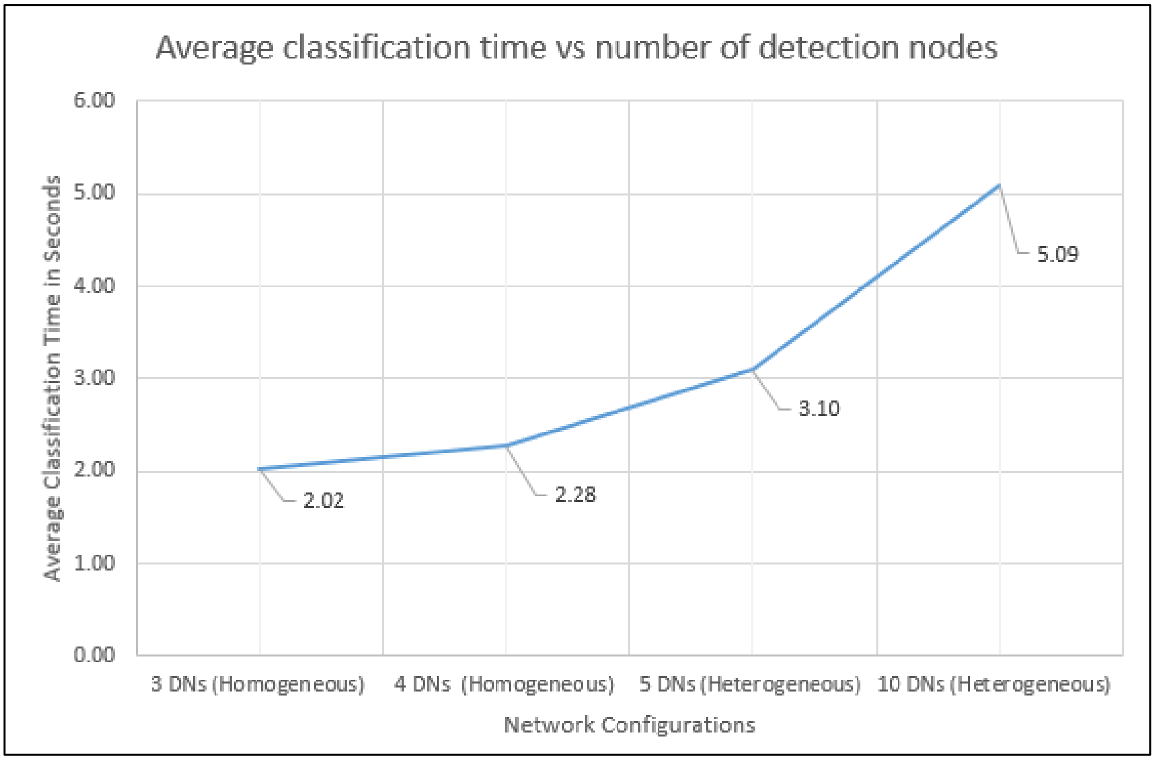


Figure 4-11: Average classification time for test set vs network configurations.

4.3.2.1 Performance improvement in detection node algorithm

The study investigated the reason behind this behavior related to the classification time. It was noticed that, for each tweet, the detection node was reloading the list of detection nodes to collect opinions from, and was verifying if, the connection exists between them. This way it was able to ensure the reachable detection nodes in the network. However, this was an overhead that was causing the classification time to increase as number of nodes in the network increased.

Therefore, the detection node algorithm was modified. Instead of reloading the detection nodes list and performing connection verifications for each tweet, these operations were performed at the initialization of a detection node.

The design of experiments in table 4-42 were repeated to observe the effect of the above modifications. The table below shows comparisons between classification times recorded with and without modification made to detection node algorithm:

Table 4-44: Comparison between classification time recorded w/ and w/o performance improvement added to code – Ten nodes heterogeneous network.

Average	Opinions	Merging	Classification Time w/o Performance improvement (s)	Classification Time w/ Performance improvement (s)
Reference	NA	NA	0.62	0.62
Case 1	2	OR	5.40	1.33
Case 2	3	OR	5.08	1.26
Case 3	2	AND	5.35	1.32
Case 4	3	AND	5.08	1.24
Case 5	2	Majority	4.36	1.25
Case 6	3	Majority	5.25	1.27

4.4 Failures in Distributed Systems

The distributed collaborative approach proposed in this research may suffer from failures in the system, as partial failures are inherent to any distributed system. In this section, we have discussed how to make the distributed collaborate approach fault tolerant. The system is fault tolerant if it continues normal operations and provides services in the presence of failure of some of its components (Fault tolerance , 2017).

The collaborative approach involves a number of detection nodes and communication between them. Thus it may experience failures such as crash failure (a node going down), or communication failures (slow response from nodes). It is important for the system to devise a strategy in cases when the system encounters failure. The first part of this strategy includes an immediate action, and the second part includes recovery or reversion.

The following are strategies that can be considered in case of failures in the system:

- One strategy to handle the failures could be simply to ignore them. If an entry point sends a tweet to other detection nodes to seek their assessments of the tweet, and one of these nodes goes down while processing the tweet or timeout is reached before the tweet reaches the entry point, then, the entry point ignores all the assessments it has received. Before the next tweet is sent for classification, the entry point updates the list of detection nodes to take opinions from. This process of updating the list of reachable detection nodes allows the system to isolate faults. However, there is no "recovery" mechanism provided in this strategy.

- The second strategy to handle failures could be to continue with the assessments received and the entry point makes classification decisions depending on the currently received opinions. It does not make an attempt to get the required number of opinions. In this strategy, although the system is operational, it is finding a work-around and normal operations are modified to save time.
- The third strategy could be to re-do the task with other detection nodes. Once the system encounters a failure, it goes into the recovery mode and updates the list of reachable detection nodes. Then it will select different detection nodes to take opinions from and re-do the task it was originally performing for the given tweet. The system can define how many such failures it can handle.

We have implemented the third strategy discussed above and tested it in the Case 5 experiment performed in Section 4.3.2.1 that has 10 detection nodes system with Majority merging technique.

Experiments were performed with the same set up as in Section 4.3.2.1. Additionally, to validate the behavior with failures, after starting the experiment with all 10 nodes in normal operating mode, a single node selected at random was intentionally stopped to emulate a detection node going down in the network. Then another node selected at random was intentionally stopped to emulate multiple nodes going down in the network. The results of these experiments are shown in the table below.

Table 4-45: Collaboration Case 5 performance DN10 is acting as entry point – Ten nodes heterogeneous network with single node failure.

Average	Opinions	Merging	Accuracy (%)	Precision (%)	Recall (%)	Time (s)
Case 5 w/o failures	2	Majority	69.01	44.88	42.15	1.24
Case 5 w/ single failure	2	Majority	67.03	43.62	58.08	1.31
Case 5 w/ two failures	2	Majority	66.90	43.08	53.45	1.26

It can be noted that the classification time with failure(s) in the system has not drastically increased.

This concludes the experiments for the distributed collaborative approach for cyberbullying detection. The experiments performed with three, four, and five detection nodes for homogeneous networks provide important insights into the effect of parameters such as number of opinions, merge technique, and opinion condition on results of the collaboration. This will allow for building better configurations for collaboration. The experiments performed with heterogeneous networks where detection nodes use different classification algorithms bolster the need of collaboration. Additionally, the experiments performed with ten detection nodes establish that the proposed approach is scalable and can work in an environment with a large number of detection nodes. The next section examines if this approach can be generalized and used for other domains such as politics.

4.5 Detection of Political Tweets

In this section, we discuss the possibility of using the approach in other domains such as politics. The training dataset and model generation code that is used to train classifiers using Naive Bayes, Logistics, and Support Vector Machine were borrowed from a recent PhD thesis in our group (Gamage, 2016).

The test dataset for the experiments was created by scraping the tweets from Twitter itself. There are a total of 100 tweets in the test dataset. It has 30 non-politics related tweets and 70 politics related tweets.

The entire political dataset collected for the previously mentioned work was used for training each of the machine learning algorithms. The experimental setup had 3 detection nodes, each detection node was using a model trained on one of the three algorithms mentioned above. In order to study the performance of each detection node, the test set was classified by each detection node individually. The classification performance of each node is indicated in the Table 4-46. Parameters used for the entry point in this experiment are listed in Table 4-47.

Table 4-46: Performance of individual detection nodes in the network – Political tweet detection.

Node	Accuracy (Percentage)	Precision (Percentage)	Recall (Percentage)	Time (seconds)
DN1 (SVM)	76.00	87.50	23.33	0.49
DN2 (Logistics)	33.00	23.94	56.67	0.31
DN3 (Naive Bayes)	34.00	20.00	40.00	0.52

Table 4-47: Fixed parameter values for Case 1 to Case 4 – Political tweet detection

Parameter Name	Parameter Value
host	DN3 (Naive Bayes)
isEntryPoint	True for DN3
Model	Entire Political Tweet Dataset
Algorithm	Naive Bayes
detectionNodeList	DN1 (SVM) and DN2 (Logistics)
selectionType	Random
opinionCondition	Classified as non-bullying

Table 4-48: Collaboration cases performance when DN3 is acting as entry point – Political tweet detection

Average	Opinions	Merging	Accuracy (%)	Precision (%)	Recall (%)	Time (s)
Ref.	NA	NA	34.00	20.00	40.00	0.52
Case 1	One	OR	35.20	27.78	72.67	0.60
Case 2	Two	OR	34.00	30.43	93.33	0.54
Case 3	Two	AND	35.00	21.31	43.33	0.61
Case 4	Two	Majority	35.00	21.31	43.33	0.59

The results of the experiments are indicated above in Table 4-48. It can be seen from Table 4-47 that collaboration cases have higher precision and recall compared to the no-collaboration case. The ‘OR’ merging technique turns out to be a clear winner in the collaborative scenario. It improves the recall of the bad performing node by 32% with the addition of a single opinion. True positives detected by other nodes are added to the result. Naive Bayes training model is able to correctly identify 12 tweets out of 30 political tweets in the political dataset. Analysis of a single run of the Case 1 experiment shows that collaboration adds 9 more true positives to the results, which brings the total true positive to 23. For Case 2, 28 out of 30 political tweets are identified by the detection node with collaboration. Precision is also improved in both cases due to the addition of true positives. ‘AND’ and ‘Majority’ merging techniques were not able to improve the performance drastically due to the fact that each machine learning algorithm was able to identify a different set of political tweets from the test set and these two merging techniques expect consensus between detection nodes.

The results obtained for political domain are consistent with the results obtained for the Cyberbullying domain. The distributed-collaborative approach improves precision and recall of a detection node in the network for other domains as well. From these experiments, we can infer that the proposed collaborative solution can be generalized for other domains as well.

5. SUMMARY AND CONCLUSION

This thesis has provided a collaborative approach for detecting cyberbullying in tweets using different distributed collaboration patterns. Various experiments carried out indicate that the collaborative approach performs better than the stand-alone approach. The following are the contributions of the thesis.

- Distributed-Collaborative approach was tested using experiments that were performed with three, four, and five detection nodes networks that has homogeneous configurations.
- The approach was validated for use in heterogeneous network configurations with five and ten nodes.
- Distributed-Collaborative approach was tested in presence of failures and various strategies were proposed.
- Distributed-Collaborative approach was tested on other domain such as politics.
- From the results of the three, four, and five detection nodes studies performed in Section 4.2, it can be concluded that the 'OR' merging technique with 2 or 3 opinions form an optimum configuration for distributed collaborative approach as it yields better recall in all the cases as compared to 'AND' and 'Majority' techniques.
- The 'AND' merging technique fails to provide competitive recall values if the network contains a bad performing node.
- Similar results are obtained for homogenous and heterogeneous network configurations.

Many future extensions of this work are possible. Some of these include:

- Examine the possibility of selecting a node depending on the opinion provided for the past tweets. This type of selection technique will be heuristics based selection. In this case, detection nodes may use reinforcement learning to define which nodes are providing better opinions than others. Initially, all the detection nodes in network will have same weight, with time only better opinion providing nodes will be contacted for opinions. This will allow us to prune out bad performing nodes and increase performance.

- Examine the network behavior when each detection node is running one of the proposed approaches in literature.
- Exploring other merging techniques such as changing majority condition to classify a tweet as bullying if half of the opinions collected classify the tweet as bullying.
- Exploring the possibility of using ‘Ensembles of classifiers’ (Ensembles of classifiers, 2016). Ensembles of classifiers combine classifiers to improve performance of single classifier. The result integration algorithms tested with this proposed technique can be explored with the distributed-collaborative approach.

REFERENCES

"Cyber Bullying Statistics". (2017). Retrieved from https://www.guardchild.com/: https://www.guardchild.com/cyber-bullying-statistics/

"How to Report". (2017). Retrieved from http://www.proudlysa.biz/: http://www.proudlysa.biz/stopbullying/index.php/stop-bullying/cyber-bullying/129-stop-bullying-how-to-report-bullying-or-abuse-on-social-media

"Learn how to control your Twitter experience". (2017). Retrieved from https://support.twitter.com/: https://support.twitter.com/articles/20170134

Algar, S. (2017, 02 01). *Cyberbullying in city schools soars 351% in just two years*. Retrieved from http://nypost.com/: http://nypost.com/2017/02/01/cyberbullying-in-city-schools-soars-351-in-just-two-years/

Basic Privacy Settings & Tools. (2017). Retrieved from https://www.facebook.com/: https://www.facebook.com/help/325807937506242/

Bosse, T., & Stam, S. (2011). A normative agent system to prevent cyberbullying. *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference* (pp. Vol. 2, pp. 425-430). IEEE.

Cross-validation (statistics). (2017). Retrieved from https://en.wikipedia.org/wiki/: https://en.wikipedia.org/wiki/Cross-validation_(statistics)

Cyber Bullying On The Rise: Facebook, Ask.Fm And Twitter The Most Likely Sources. (2015). Retrieved from http://www.huffingtonpost.co.uk: http://www.huffingtonpost.co.uk/2014/08/14/cyber-bullying-on-the-rise-facebook-ask-fm-and-twitter-the-most-likely-sources_n_7359146.html

CyberPatrol Parental Controls 7.7. (2017). Retrieved from https://www.amazon.com/: https://www.amazon.com/CyberPatrol-CPC-1-12-N-SBXRT-Parental-Controls-7-7/dp/B001KF284U/

CyberPatrol Parental Controls 7.7. (2008). Retrieved from http://www.pcmag.com/: http://www.pcmag.com/article2/0,2817,2334007,00.asp

Chen, Y., Zhou, Y., Zhu, S., and Xu, H. (2012). Detecting offensive language in social media to protect adolescent online safety. *Privacy, Security, Risk and Trust (PASSAT) 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom)*, (pp. 71-80).

Dinakar, K., Jones, B., Havasi, C., Lieberman, H., & Picard, R. (2012). Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *2*(3), 18.

Dinakar, K., Reichart, R., & Lieberman, H. (2011). Modeling the detection of Textual Cyberbullying. *The Social Mobile Web*, 11(02).

Email and Web Security with Data Loss Prevention. (2014). Retrieved from https://www.watchguard.com: https://www.watchguard.com/docs/datasheet/wg_xcs_product_matrix.pdf

Ensembles of classifiers. (2016, October 28). Retrieved from https://en.wikipedia.org/wiki/: https://en.wikipedia.org/wiki/Ensembles_of_classifiers

Fault tolerance. (2017 , April). Retrieved from https://en.wikipedia.org: https://en.wikipedia.org/wiki/Fault_tolerance

Gamage, D. U. (2016). *QoS and trust prediction framework for composed distributed systems.* Indianapolis: Purdue University.

George-Nektarios, T. (2013). Weka classifiers summary. *Athens University of Economics and Bussiness Intracom-Telecom, Athens*.

Hinduja, S., & Patchin, J. W. (1998). Cyberbullying research summary. *Developmental Psychology*, 34(2), 299-309.

Hinduja, S., & Patchin, J. W. (2007). Offline consequences of online victimization: School violence and delinquency. *Journal of school violence*, *6*(3), 89-112.

Kasture, A. S. (2015). *A predictive model to detect online cyberbullying.* (Doctoral dissertation, Auckland University of Technology).

Kontostathis, A., Reynolds, K., Garron, A., & Edwards, L. (2013, May). Detecting cyberbullying: query terms and techniques. *Proceedings of the 5th annual acm web science conference*, (pp. 195-204). ACM.

Li, R. (2012). *Dataset: UDI-TwitterCrawl- Aug2012.* Retrieved from https://wiki.cites.illinois.edu/wiki/display/forward/Dataset-UDI- TwitterCrawl-Aug2012.

Mangaonkar, A., Hayrapetian, A., & Raje, R. (2015, May). Collaborative detection of cyberbullying behavior in Twitter data In *Electro/Information Technology (EIT), 2015 IEEE International Conference on*. (pp. 611-616). IEEE.

McQuade, S. C., Colt, J. P., & Meyer, N. B. (2009). *Cyber bullying: Protecting kids and adults from online bullies*. ABC-CLIO.

Nahar, V., Li, X., & Pang, C. (2013). An effective approach for cyberbullying detection. *Communications in Information Science and Management Engineering*, *3*(5), 238.

Ng, A. Y., & Jordan, M. I. (2002.). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems* (pp. 841-848)

Online Child Safety. (2011). Retrieved from http://www.puresight.com/: http://www.puresight.com /

Patchin, J. W. (2014, December 23). *What is Cyberbullying?* Retrieved from http://cyberbullying.org: http://cyberbullying.org/what-is-cyberbullying

Peterson, J. (2013, March 16). *Survey: Majority of cyber bullying incidents go unreported* . Retrieved from http://www.topnews.in/ : http://www.topnews.in/survey-majority-cyber-bullying-incidents-go-unreported-2375044

Reynolds, K., Kontostathis, A., & Edwards, L. (2011, December). Using machine learning to detect cyberbullying . *Machine learning and applications and workshops (ICMLA), 2011 10th International Conference* (pp. Vol. 2, pp. 241-244). IEEE.

Shipley, R. (2017). *Surfie by*. Retrieved from http://www.toptenreviews.com/: http://www.toptenreviews.com/software/security/best-internet-filter-software/puresight-pc-details/

Šléglová, V., & Cerna, A. (2011). Cyberbullying in adolescent victims: Perception and coping. *Cyberpsychology: journal of psychosocial research on cyberspace*, *5*(2).

Steinmetz, K. (2014, Feb 19). *#Cursing Study: 10 Lessons About How We Use Swear Words on Twitter.* Retrieved from time.com: http://time.com/8760/cursing-study-10-lessons-about-how-we-use-swear-words-on-twitter/

Stop Cyber-Bullying in its Tracks. (2011, January). Retrieved from https://www.watchguard.com: https://www.watchguard.com/docs/tech/wg_cyber_bullying_tb.pdf

Twitter Usage Statistics . (n.d.). Retrieved from http://www.internetlivestats.com/twitter-statistics/#rate

Tynes J. (2014, November 26). *Twitter Sentiment Algos — Benchmarking Precision, Recall, F-measures.*. Retrieved from https://www.linkedin.com/: https://www.linkedin.com/pulse/20141126005504-34768479-twitter-sentiment-algos-benchmarking-precision-recall-f-measures

Waikato, T. U. (n.d.). *Weka 3: Data Mining Software in Java* . Retrieved from http://www.cs.waikato.ac.nz/ml/weka/.

WatchGuard Firebox. (2014). Retrieved from http://www.pcmag.com/: http://www.pcmag.com/article2/0,2817,2462437,00.asp

Waugh, R. . (2014). *Half of children left exposed to online threats as parents fail to use built-in controls*. Retrieved from https://www.welivesecurity.com/: https://www.welivesecurity.com/2014/02/11/half-of-children-left-exposed-to-online-threats-as-parents-fail-to-use-built-in-controls/

What is Cyberbullying. (n.d.). Retrieved from http://www.stopbullying.gov/: http://www.stopbullying.gov/cyberbullying/what-is-it/index.html

Why do kids cyberbully each other?. (n.d.). Retrieved from http://www.stopcyberbullying.org: http://www.stopcyberbullying.org/why_do_kids_cyberbully_each_other.html

Why Do People Cyberbully?. (n.d.). Retrieved from http://endcyberbullying.net/: http://endcyberbullying.net/why-do-people-cyberbully/

Worried about your child’s phone or tablet usage?. (2017). Retrieved from http://www.phonesheriff.com/: http://www.phonesheriff.com/

Yin, D., Xue, Z., Hong, L., Davison, B. D., Kontostathis, A., & Edwards, L. . (2009). Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB*, (pp. 2, 1-7.).

Zhang, X., Tong, J., Vishwamitra, N., Whittaker, E., Mazer, J. P., Kowalski, R., Hu, H., Luo, F., Macbeth, J. & Dillon, E. (2016 December). Cyberbullying Detection with a Pronunciation Based Convolutional Neural Network. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on* (pp. 740-745). IEEE.

VITA

Amrita Mangaonkar, computer science graduate student with Indiana University Purdue University of Indianapolis. Currently working with Synopsys, Inc. as security consultant at Bloomington, Indiana.