MULTILINGUAL CYBERBULLYING DETECTION SYSTEM

by

Rohit S. Pawar

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 Science

Indianapolis, Indiana

May 2019

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Rajeev R. Raje, Chair

Department of Computer and Information Science

Dr. Mihran Tuceryan

Department of Computer and Information Science

Dr. Arjan Durresi

Department of Computer and Information Science

Approved by:

Dr. Shiaofen Fang

Head of the Graduate Program

Dedicated to loving memory of my grandparents.

ACKNOWLEDGMENTS

It gives me great pleasure in presenting this thesis report titled: “MULTILINGUAL CYBERBULLYING DETECTION SYSTEM”. I express gratitude to my Advisor, Dr. Rajeev R. Raje, who provided me with all the guidance and encouragement and making the lab available to me at any time. I also would like to deeply express my sincere gratitude to my Advisory Committee members, Dr. Mihran Tuceryan and Dr. Arjan Durresi, for their insightful comments and encouragement, which encouraged me to widen my research from various perspectives. Additionally, I would like to thank Dr. Pushpak Bhattacharyya from IIT Bombay for providing the required datasets to carry out the research.

I would like to thank Ms. Nicole Wittlief for providing her valuable input on my dissertation. I am eager and glad to express my gratitude to my colleagues Amrita, Yash and Akshay for the assistance, detailed suggestions and also encouragement to do the project. Special thanks to Amrita Mangaonkar for letting me use her work for the validation of my work.

I am extremely thankful to IUPUI, and all staff and the management of Department of Computer and Information Science for offering me the Dean’s Scholarship that allowed me to carry out my research.

Finally, yet importantly, I would like to thank my entire family for substantial support throughout my master's journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Author: Pawar, Rohit, S. MS
Institution: Purdue University
Degree Received: May 2019
Title: Multilingual Cyberbullying Detection System
Committee Chair: Rajeev R. Raje

Since the use of social media has evolved, the ability of its users to bully others has increased. One of the prevalent forms of bullying is Cyberbullying, which occurs on the social media sites such as Facebook©, WhatsApp©, and Twitter©. The past decade has witnessed a growth in cyberbullying – is a form of bullying that occurs virtually by the use of electronic devices, such as messaging, e-mail, online gaming, social media, or through images or mails sent to a mobile. This bullying is not only limited to English language and occurs in other languages. Hence, it is of the utmost importance to detect cyberbullying in multiple languages. Since current approaches to identify cyberbullying are mostly focused on English language texts, this thesis proposes a new approach (called Multilingual Cyberbullying Detection System) for the detection of cyberbullying in multiple languages (English, Hindi, and Marathi). It uses two techniques, namely, Machine Learning-based and Lexicon-based, to classify the input data as bullying or non-bullying. The aim of this research is to not only detect cyberbullying but also provide a distributed infrastructure to detect bullying. We have developed multiple prototypes (standalone, collaborative, and cloud-based) and carried out experiments with them to detect cyberbullying on different datasets from multiple languages. The outcomes of our experiments show that the machine-learning model outperforms the lexicon-based model in all the languages. In addition, the results of our experiments show that collaboration techniques can help to improve the accuracy of a poor-performing node in the system. Finally, we show that the cloud-based configurations performed better than the local configurations.

CHAPTER 1. INTRODUCTION

With the advent of internet and technology, social media has emerged as a major part of our life. It helps us keep in touch with one another with the use of different applications with just a few taps and/or swipes. It is a constant source of entertainment. People have started feeling more sociable despite their current situation, even if they are at home or at work. With our smartphones and tablets the social media platforms are easily accessible, there has been a rise in the number of users over the past few years. The global digital report created by Dave Chaffey in 2018 [1] indicates the following statistics related to internet user – there are around 4.021 billion Internet users, 3.196 billion social media users and 5.135 billion mobile phone users. However, social media has its own difficulties and challenges. For example, social media may contain a lot of antisocial behavior, including cyberbullying, cyber stalking, and cyber harassment. These behaviors have now become part our lives and are not only bounded to juveniles, but any person can be a victim of it.

1.1 Cyberbullying

Cyberbullying is an oppression happening virtually using devices such as computers, mobiles, and tablets. Cyberbullying can take place through messaging or on the internet in forums, social platforms, or gaming where community can share and post their thoughts. In short, social media are being used by bullies to harass people. Bullying can be analysed by consecutive behaviour and a purpose to harm which leads to having suicidal ideation, emotional responses and lower self-esteem such as the victim being angry, frustrated, scared and depressed.

Children, in today’s world, want their own mobiles and tablets at an adolescent age, and desire to connect to the social media platforms, and play online games such as Fortnite. If their behaviour goes unmonitored by their parents, then it may lead to cyber bullying.

Examples of cyberbullying can include rumors posted on social media or sent by e-mail; embarrassing videos or pictures; and insulting, intimidating and abusive messages posted on social networks. When a message or a picture is posted online, it is very difficult to track and remove the content from the social media. This can take place 24x7, and it can stretch out to its victim when they are away from home, alone [2]. Cyberbullying is a bit contrasting from traditional bullying as the offender does not have to physically tackle their victims.

Some of the prominent definitions of cyberbullying are:

- “An aggressive, intentional act carried out by a group or individual, using electronic forms of contact, repeatedly and over time against a victim who cannot easily defend him or herself” [3].
- “Cyberbullying is when someone repeatedly makes fun of another person online or repeatedly picks on another person through e-mail or text messages or when someone posts something online about another person that they don’t like” [4].

Cyberbullying can include sending mean messages or DMs to someone, pranking peoples calls, harassing someone in an online game, hacking into someone’s social networking profile or game, spreading rumors about people online, and pretending to be someone else to spread hurtful messages online.

1.2 Effects of Cyber-bullying

Traditional bullying or cyberbullying causes psychological and emotional distress. In fact, similar to any other victims of bullying, cyberbullied kids or teenagers experience depression, fear, low self-esteem, and anxiety. They also may experience physical symptoms, and academic struggles. In addition, the victims of cyberbullying also experience some consequences and feelings. These are:

- Elevated feelings of isolation, sadness, and anxiety leading to Depression.
- Skipping or dropping out of school
- Health related criticisms
- Depreciated academic grades, intellectual accomplishments, and standardized exam scores and school involvement
- Variations in eating and sleeping patterns, and lack of interest in hobbies and habitual activities.

The above mentioned may even lead to suicidal tendencies.

1.3 Countermeasures by Social Media

Recent statistics obtained from various sources show the following:

- "Youths experienced cyberbullying on Instagram more than any other platform (at 42%), with Facebook following close behind (at 37%) and Snapchat ranked third (at 31%). While the surveyed participants, in a study, used YouTube more than any other platform, the video-focused social media was only responsible for 10% of the reported cyber bullying. 71% of the survey participants said that social media platforms do not do enough to prevent cyberbullying" [5].
- "A 2016 report from the Cyberbullying Research Center indicates that 11.5% of students between 12 and 17 indicated that they had engaged in cyberbullying in their lifetime. Conversely, 33.8% of students between the case of 12 and 17 were victims of cyberbullying in their lifetime. Conversely" [6].
- "In a random sample study over 14% admitted to cyberbullying another person, with spreading rumours online, via text, or email being the most common form of bullying" [7].
- "A study by McAfee, found that 87% of teens have observed cyberbullying" [8].
- "54% of teens surveyed have witnessed online bullying" [9]:
    - 39% on Facebook
    - 29% on YouTube
    - 22% on Twitter
    - 22% on Instagram

Social networks and other services provide an extent of support for a protected web experience. The following tools are helpful for providing protection to one's privacy:

- Twitter© endeavors to provide a space for people to express themselves freely. Additionally, provides a medium for users to report any sort of abusive content on the platform. The user can include multiple tweets in the same report, helping Twitter to improve its context, while inspecting the problems to get them resolved sooner. In addition, a user can block, mute or unfollow other unwanted users.

- Facebook ® has partnered with the Yale Center to develop a bullying prevention hub for Emotional Intelligence for users seeking support and aid for issues associated to bullying and additional conflicts. The hub offers systematic plans, like guiding how to start important conversations for people being bullied, for parent whose children where bullied or suspected of bullying, and educators who have had students involved with bullying [10].
- Instagram provides a feature to block users who post offensive or inappropriate behavior.
- Stopbullying.gov offers a full guide to forthcoming cyberbullying and is presented by the U.S. Department of Health and Human Services [11].
- The updated site of the Cyberbullying Research Center offers information on the laws for individual state, along with how those laws might assist with reporting, blocking or otherwise ending the harassment [6][7].

1.4 Need for Detection of Cyberbullying and Motivation of the Research

All the methods used by the social media platforms, such as Facebook and Twitter, utilize filter after a post has already been made – i.e., these are posterior actions. Simultaneously, there is no system (outside of our group [32][35]), that is available in present for automatic detection of such behavior so, many users can still view most of these posts (unless the posts are flagged and reported).

Cyberbullying is not location and language specific, i.e., it occurs worldwide and across different languages. Since cyberbullying, occurring anywhere and in any language, can have long-lasting effects on the victims, an automatic detection system that can detect cyberbullying posts in different languages should be in place. Such a system can help showing a warning message to the sender. Most of the prevalent approaches to automatically detect cyberbullying (indicated in Chapter 2) focus on English text and associated forums. However, multiple mobile device users are in Asian countries such as India, Japan, China and South Korea [13]. For example, in India, there are 1.16 billion mobile device users [13] and they are very active on various social media forums such as WhatsApp and use the Indian languages and their features associated with such apps. This sheer volume of users necessitates the creation of an automatic cyberbullying detection system in other languages. This will help the victims of cyber bullying around the world, and such a system will be able to monitor and filter the hateful, improper, abusive content from social media posts.

1.5 Problem Definition

This thesis describes a Multilingual Cyberbullying Detection System for detection of cyberbullying behavior in English and two Indian languages – Hindi and Marathi. These two languages have 293 million (4.46% of world's population) and 73 million (1.1% of world's population) native speakers [14]. Hence, the proposed system has the potential to have a significant impact in making online forums safer for the users of these two languages. The specific objectives of this thesis are:

- To design, implement and experiment with a Multi-Lingual (Hindi, Marathi and English) Cyberbullying Detection System that uses different machine-learning algorithms.
- To incorporate various distributed techniques into the proposed system and study their consequences on the time and the precision of detection of cyberbullying content.

1.6 Overview of the Proposed Approach

Various approaches, in recent past, have been proposed to detect cyberbullying in textual content (refer chapter 2). These include Machine-learning (ML) techniques and Natural Language Processing (NLP) techniques. Many of the ML techniques involve supervised learning, while the NLP techniques include Bag-of-Word (BoW) and Lexical Syntactical Features (LSF). In addition, most of the prevalent approaches are mainly sequential in nature and do not consider distribution, which is an inherent feature of cyberbullying behavior. Hence, our proposed system uses two types of distributed system architectures. First, we have created a local distributed infrastructure (discussed in 2.4). However, maintaining this infrastructure was a challenge (due to auto-scaling, handling fail-over in case of all node fails, automatic spinning up new node, etc.). Hence, we decided to use the cloud and shift these challenges to the cloud providers. We also created another prototype of our system using Amazon's cloud service, i.e., AWS Elastic Beanstalk [15]. We have created a REST end point and deployed it into the AWS cloud so that any client can make an API call and get prediction results in return. This service acts as a third-party service so that any application can call this service from their application.

1.7 Evaluation

The success of this thesis is measured using the following metrics:

- An effective and efficient detection of cyberbullying in text written in multiple languages and across multiple datasets.
- An ability to detect multi-lingual cyberbullying in real-time and the prevention of the delivery of bullying messages to other users.
- An incorporation of fault tolerance in the system and associated scalability study.

Different experiments have been carried out with the prototype (described in Chapter 4) to address these metrics.

1.8 Contributions

The contributions of this thesis are as follows:

- It proposes a system which can detect cyberbullying in multiple languages, as evident from empirical evaluation.
- It implements a system which can scale easily and provide a fault tolerant capability.

1.9 Organization

This thesis content is composed of five chapters. The first chapter comprises of an introduction accompanied by the motivation and problem definition. The second chapter discusses related work in this domain. The third chapter provides the system architectures of the proposed system. Furthermore, it discusses metrics used to validate the proposed approach. The fourth chapter analyses the results of the empirical estimations with the prototypes. Finally, chapter 5 indicates the conclusion and future work.

CHAPTER 2. RELATED WORK

This chapter discusses the problem of cyberbullying and presents an overview of previous work done by others in this domain to detect bullying.

2.1 Research Related to Cyberbullying Detection

Many researchers have studied and analyzed effects of cyberbullying on society. “The emotional and psychological consequences of cyberbullying have been broadly studied by Hinduja et al. in” [16]. Many surveys have been carried out each year to study the nature of cyberbullying and generate guidelines that can benefit victims to accord with the problem [6][7].

In some instances, the expert guidance is to keep a watch on children’s social media activity [17]. Although these approaches seem useful, they are lacking to address the overall problem of automatic cyberbullying detection, as children report minimal number of incidents to their parents or social media directly [18]. Moreover, regardless of these procedures, cyberbullying behaviour is growing in today’s world [19].

2.2 Content Filtering Software

There are many content filtering software systems available in the market including BullyBlocker [20], SafeChat [21], Facebook WatchDog [22], and Rethink [23]. These software systems can be used by parents to monitor the social media activity of children and thus can help detect and prevent cyberbullying. A few instances of such software systems are listed below:

BullyBlocker: They created a computational model to detect and compute the intensity of cyberbullying in social media platforms. This app detects cyberbullying on Facebook and notifies a parent/ teacher when cyberbullying occurs.

SafeChat: This is a software tool for Facebook, which protects children’s communication from explicit messages. They have developed a model which look for specific words in communication and drops such messages.

Facebook WatchDog: This software detects threats and cyberbullying on social media. They detect threats using social media analytics, image analysis, and text mining techniques.

Rethink: This app can be installed on any device. As soon as a user types some word, this app runs in the background and looks for abusive word. It then shows a warning message, if that word belongs to bullying category.

Although these software can provide a method for parents to keep an eye on children's activity, but it often fails to fully exploit the potential of such software as it can be easily bypassed by witty children [24].

Further, there is no collaboration between the different software of similar type. In addition, parent tends to fail to utilize the software, so such instances go unpunished and without reported to trusted authority.

2.3 Cyberbullying Detection Techniques

Cyberbullying detection approaches mainly fall under two categories: machine learning techniques and lexicon-based techniques.

2.3.1 Machine Learning Techniques

In this approach, different supervised learning and unsupervised learning algorithms are used to detect cyberbullying. Common steps followed in this approach are: gathering a dataset and tagging it, preprocessing on the dataset, training the machine learning model and testing it on some portion of the dataset. Here we describe prominent approaches that mainly deal with multiple languages.

In [25], Haider et al. conducted a survey on multilingual cyberbullying detection. In the survey, they found that most of the work in this domain is focused on English texts. They attempted cyberbullying detection in the Arabic language in [26]. In their work, they used the ML learning approach to detect cyberbullying. Their dataset contains 32K tweets out of which 1800 tweets were bullying ones. They used Support Vector Machine (SVM) and Naïve Bayes (NB) algorithms to detect cyberbullying and got 92% and 90% F1 scores respectively.

Ting et al., in [27], gathered a dataset from 4 popular social sites in Taiwan. They used Social Network Mining (SNM) technique to detect cyberbullying. In this, they identify three features from the data: Keywords, Social Network Analysis, and Sentiment. They have identified that sentiment is the most important feature to detect cyberbullying, as it helps to understand the intent of a user when he posts messages on social media. They used precision and recall as performance metrics and their results show the precision and accuracy are around 0.79 and the recall is 0.71.

In [28], Noviantho et al. obtained a dataset from Kaggle, which contains ~12K tweets out of which 1068 were bullying ones. In their approach, they used machine learning techniques such as SVM and naïves along with N gram technique. They used NB and SVM techniques and then applied the n-gram technique with values from 1 to 5. They observed that 92.81% of accuracy was achieved using NBs; while it was observed that 97.11% accuracy is yielded using the SVM with a poly kernel.

In [29], Silva et al. developed a mobile app called 'BullyBlocker'. The main aim of their work was to develop a mobile app on top of the machine-learning model. This app not only helps in cyberbullying detection but also sends bullying detection alerts to parents. This app scrawls the Facebook feed and messages using Facebook's API and retains a record of bullying for the last 60 days.

Nurrahmi et al., in [30], proposed a model, that not only detects cyberbullying but also keeps track of bullying between users, which can help to determine the credibility of the user. Their aim is to detect cyberbullying for the Indonesian Language. For this, they gathered around 700 tweets out of which 300 were bullying ones. They developed a machine-learning model using SVM and got an F-1 score of 67%. To determine a user's credibility, they keep track of number of bullying tweets and non-bullying tweets sent by that user. Based on this data, they calculated the user's behavior and then classified it as a bullying actor if the abnormal behavior is >60%.

In [31], Özel et al. gathered data from Twitter and Instagram written in Turkish and then applied decision tree (C4.5), SVM, Multinomial Naïve Bayes, and K Nearest Neighbors (KNN) to detect cyberbullying. As per observation, accuracy improves when they considered both words and emoticons in the text messages as features. Naïve Bayes outperformed all other algorithms in their experiment, and it achieved 84% accuracy.

In [32], Mangaonkar et al. proposed a model which uses different collaboration patterns to detect cyberbullying. They identified that the many bullying detection approaches are mostly stand-alone and hence, are inefficient while dealing with a large volume of messages. Their system is distributed in nature and hence, their system enhances accuracy and time over the stand-alone approach.

All the above-mentioned related work using machine-learning techniques have these limitations: i) Most of the work is done in only the English language, and ii) Machine-learning approaches do not take language inputs such as grammar and negation handling into the consideration. It just simply counts the occurrence of words and assign weights to words based on it.

2.3.2 Lexicon based techniques

These methods are based on the simple Bag-of-Words (BoW) technique. In this approach, a corpus of delicate, abusive, and unpleasant words is created. At that point, algorithms use this corpus to check the occurrences of these words in messages to detect bullying.

In [33], Chen et al. presented a method called Lexical Syntactic Feature (LSF) for detecting cyberbullying. LSF highly relies on BoW, for message-level abuse recognition. They achieved a precision of 98.24% and recall of 94.34% in sentence level abuse detection. In [34], Kontostathis et al. have analyzed a certain set of words that are used by cyberbullies and their context. These words are then used to form a query which can analyze cyberbullying.

All these lexicon-based techniques have the following limitations: i) they depend on the dictionary of words and weights associated with it; and ii) people use slang language/misspell the word while texting and those words might not appear in the dictionary and hence, chances are high that score of the entire text message change to exact opposite side.

2.4 Past work

2.4.1 Cyberbullying Detection Approach using Distributed Paradigm

All the approaches (except [32]) mentioned in the previous sections are sequential in nature. Social media sites such Twitter© and Facebook® generate lots of data asynchronously across globe. These social media platforms generate lots of data simultaneously and continuously from different origins. Linear approach fails to handle such a large volume of data. Hence, we need a system which not only work efficiently but also, scale and provide fault tolerant capabilities in a distributed environment. In [32], Mangaonkar et al. proposed a distributed collaborative detection system which works in such situations. Similarly, we have carried out experiments with different server configurations, in [35], which work well in a distributed environment – subsequent paragraphs in this section discuss our past work in detail.

To detect bullying as well as to allow multiple users to communicate with each other through the Communication Server, in [35] we have designed a Cyber-Bullying Detection system that enhances a chat application using socket programming in Python. The system architecture is shown in Fig 2.1. Our system has 3 main components:

- Communication Server
- Bullying Detection Server
- Chat Service

We discuss each of them in detail below.

1) Communication Server: The Communication Server is responsible for the following tasks:

- To accept multiple incoming connections from the users.
- To read incoming messages from a particular user and deliver them to the intended user(s) or broadcasts them to all other connected users, in the case of group communication.
- To forward a message to the Bullying Detection Server and take decision about forwarding/dropping the message based on the server's response.
- To take over the bullying detection activity in the case of a crash of the Bullying Server.

2) Bullying Detection Server: The Bullying Detection Server is responsible for the following tasks:

- To listen for incoming messages from the Communication Server.
- To run the bullying detection algorithm (i.e., SGD and MNB) and send a response back to the Communication Server.

3) Chat Service: The chat service is responsible for the following tasks:

- To check user input. If the user types in a message, then sends the message to the Communication Server.
- To listen for incoming messages from the Communication Server.

A user first sets up a connection with the Communication Server and authenticates using the proper credentials. A user can then send and receive messages with the help of the Communication Server and all the messages are encrypted/decrypted using a private key as shown in Fig 2.1.

The system has two work-flows as indicated below:

- Normal Work-flow: When there are no failures or errors in the system, the normal work flow is executed, which is represented by the solid line in Fig 2.1. Chat services are used to communicate to the Communication Server when a user intends to send a message to another user. This message is forwarded by the Communication Server to the Bullying Server and then waits for the response of the Bullying Server to decide whether to forward or drop a message. If the Bullying Server declares the message as non-bullying, then the Communication Server forwards the message to the user or else drops it.
- Fallback Work-flow: When there are failures or errors in the system the fallback work flow is executed which is represented by the dotted line in Fig 2.1. In case of failures in the Communication Server, the Chat Service connects the user to the backup server. The task of the backup server is to retrieve all the users' state information and history from the database and continue chat execution. If there is an error in the Bullying Detection Server, the Communication Server/Backup Server takes care of the bullying detection activity, which enables the system to continue operating properly in the event of the failure.

To protect all the messages from attackers and spammers, communication between all the entities in the system is encrypted using the AES algorithm as shown in Fig 2.1.

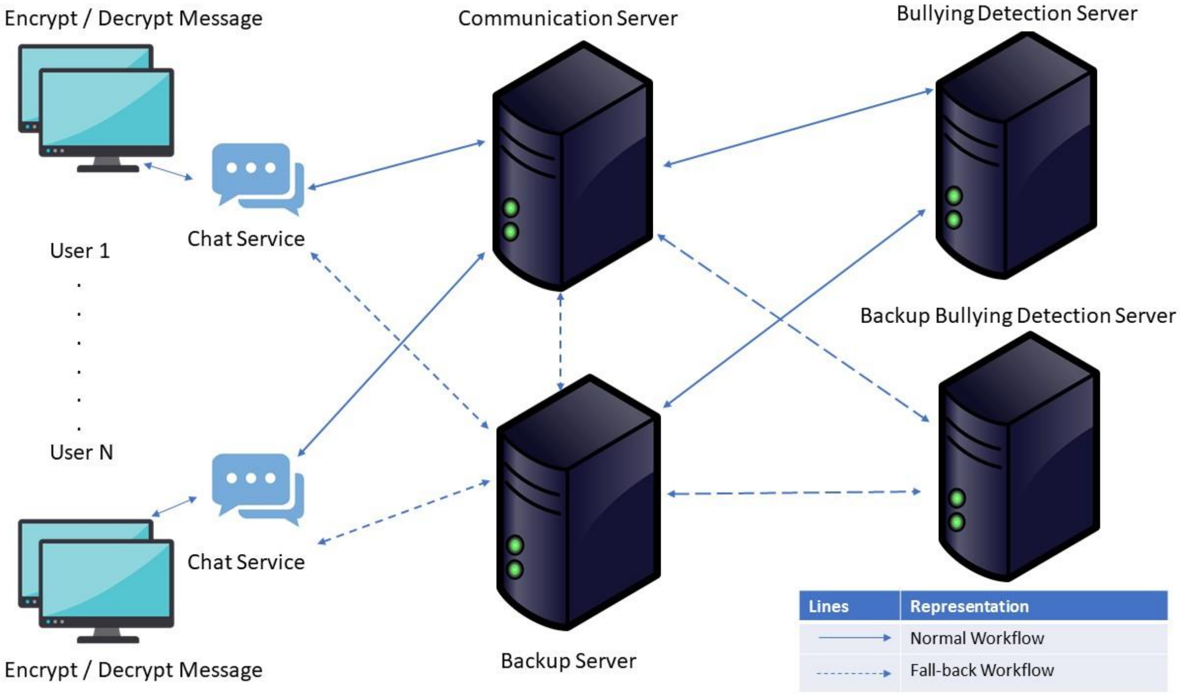


Figure 2.1: Detection System Design [35]

2.4.2 Experimental analysis with Multiple Server Configurations

In our previous work [35], we have carried out multiple experiments with different server configurations to assess the scalability and fault-tolerance. We have carried out performance analyses and the calculated average performance time of different operations with three different setups as mentioned in Table 2.1.

1) Single Server Configuration: In the Single Server configuration, a single server takes care of all the activities i.e., the communication as well as the bullying detection task. This configuration fails to achieve the fault tolerance capability. Thus, if the server fails, the entire system will fail. This configuration was the base case in our experiments.
2) Distributed Server Configuration: In the Distributed Server configuration, we have separated the functionality of the communication and bullying. Two different bullying servers have been deployed for two different algorithms. The bullying detection was performed in parallel for fast performance. For example, 2 different bullying detection algorithms on single server

required 10.364 msec, but in the case of distributed bullying the detection the time is 4.372 msec. For better accuracy, we perform the logical OR operation of results obtained from both the Bullying Servers.

3) Load Balancer Configuration: In this configuration also, we separated the communication and bullying functionality. In the case of the load balancer approach, we assign the incoming messages in a round robin fashion to balance the system workload.

We have carried out experiments using all 3 configurations and have the summarized results for each setup in Table 2.1. These results indicate that Load Balancer and Distributed Server Configurations take less time than the Single Server Configuration. The reason for this behavior is due to the fact that the Single Server Configuration has to execute multiple bullying detection algorithms on the same machine, whereas the other configurations perform the execution of algorithms on separate machines.

Table 2.1. Performance Analysis in Milliseconds

Operation	**Single Server**	**Distributed Server**	**Load Balancer**
Encryption	6.257	6.243	6.264
Decryption	0.9237	0.9123	0.9108
Communication	4.243	5.507	5.207
Bullying Detection	10.364	4.698	4.372
Total	**21.7877**	**17.0343**	**17.0798**

2.4.3 Scalability Testing

In past work [35], we have also carried out performance analysis by increasing the number of users as shown in Fig 2.2. In Fig 2.2, the X-axis represents the number of users and the Y-axis represents time in milliseconds. As seen from Fig 2.2, the end-to-end response time almost linearly increases with the number of users.

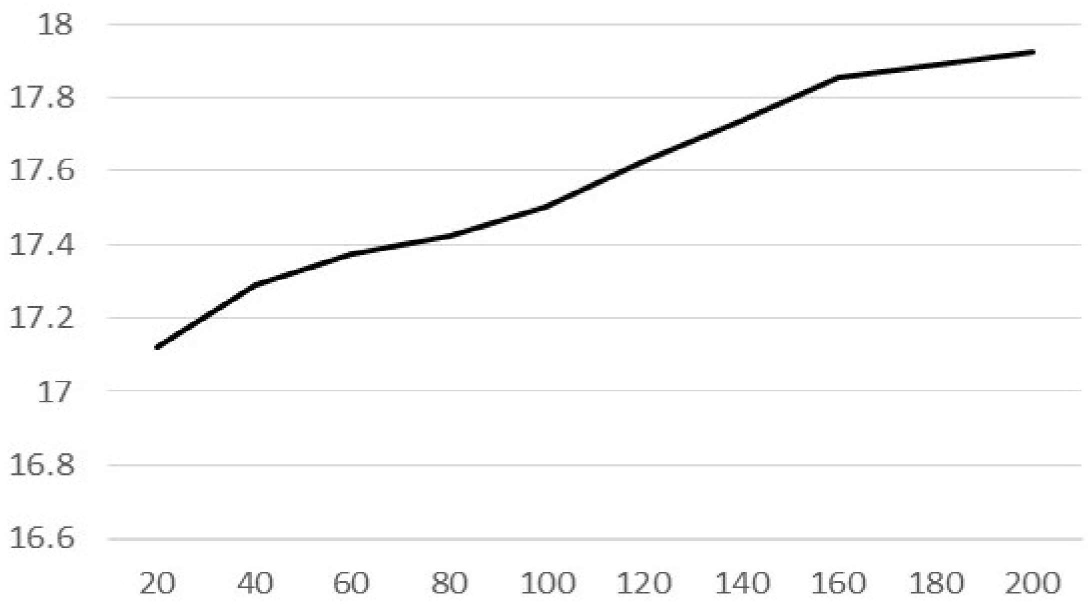


Figure 2-2. Scalability Testing for Load Balancer Configuration [35]

2.4.4 Failure Testing

We have implemented the following fault tolerance features in the system:

- If the Bullying Detection Server fails, the Communication Server takes over the functionality of the detection server and performs bullying detection.
- If one user is talking to the other user and the other user goes offline, an appropriate message is displayed to the sender.
- If the Communication Server fails, then the Backup Server takes over the communication activity.

We have done a comparative analysis of our previous prototype in [35] and a new prototype created using AWS Elastic Beanstalk [15]. We have reused parts (the chat application and the bullying detection server infrastructure) of the previous prototype for this study.

2.5 Summary

As described in this chapter, most of the work on detecting cyberbullying is performed using machine learning techniques on English text. There are very few attempts in other languages such as Arabic, Indonesian, and Turkish; however, no such efforts exist in Hindi and Marathi language texts, which is the aim of this thesis. In addition, since most of the attempts to detect cyberbullying are in stand-alone mode, when looking at inherent distributed nature of the social media data, we need a collaborative, scalable, fault tolerant system, which can achieve high accuracy and fast end-to-end response time, which is another aim of this thesis.

CHAPTER 3. PROPOSED APPROACH

In this chapter, our proposed method for multilingual cyberbullying detection is explained. The chapter also describes the data sets, the performance metrics, and the algorithms used. A prototypical system is designed, implemented, and experimented with.

3.1 Multilingual Cyberbullying Detection Approach

In this section, we discuss different architectures to detect multilingual cyberbullying. For our study, we created a chat application using Python's socket programming. The system architecture has three components. These are:

i) Chat Service: responsible for sending/receiving messages.
ii) Communication Server: responsible to maintain users' connections and to make decisions related to showing a warning message to the sender with the help of the Bullying Detection Server.
iii) Bullying Detection Server: responsible for detecting bullying behavior and returning responses back to the Communication Server.

Our approach has following assumptions:

i) We neither consider the context nor the sentiment associated with the input messages in our study. In addition, we are not addressing the problem of sarcasm detection. These issues are considered as future work.
ii) In our study, all our prediction models are developed for messages written only in one language at a time. This means that the entire input message needs to be in only one language and mixing of words from multiple languages is not acceptable. This is because, we did not find comprehensive datasets, which contain sentences written in the multiple languages. Again, such a mixed mode texts are considered as future work.

In our study, we have carried out experiments with two different setups. In the first setup, we implemented the bullying detection server using the local infrastructure. In the second setup, we used AWS Elastic Beanstalk [15] to deploy the prediction model (which is part of the Bullying Detection Server). Our proposed system has a three-tier architecture and hence, it is easy for us to replace any tier with a different setup. Overall workflow and the system design are the same as explained in Chapter 2 (refer Section 2.4.1). Below we describe the architectures associated with these two approaches. Also, in our approach, the execution of the Chat Service and the Communication Server remain the same, only the bullying detection mechanism varies in each case.

3.1.1 Distributed Cyberbullying Detection using Local Infrastructure without Collaboration

Fig. 3.1 shows the system architecture, which uses the local infrastructure. In this architecture, each bullying detection server uses multiple machine-learning (ML) algorithms (details are provided in Section 3.3 and Chapter 4) to detect cyberbullying behavior and returns a combined result to the communication server. Also, the bullying detection server executes these ML algorithms in a sequential manner.

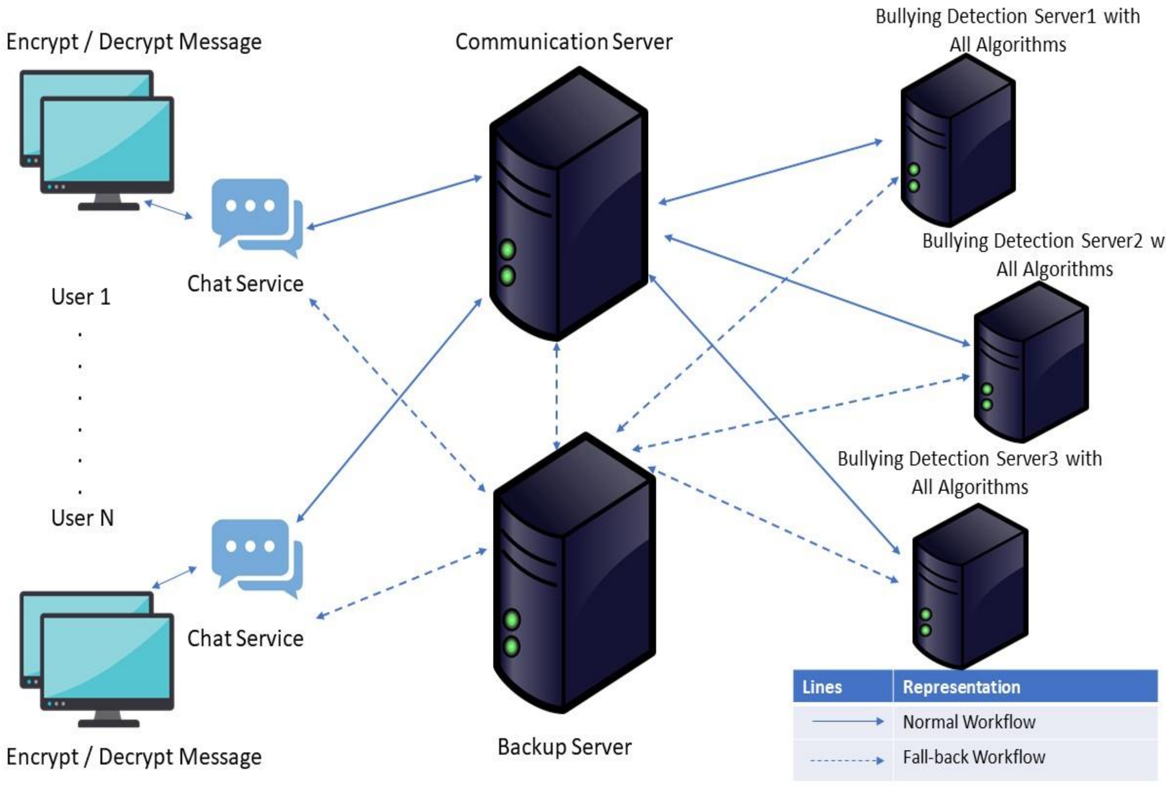

Encrypt / Decrypt Message
Communication Server
Bullying Detection Server1 with All Algorithms
Chat Service
User 1
Bullying Detection Server2 with All Algorithms
User N
Bullying Detection Server3 with All Algorithms
Chat Service
Encrypt / Decrypt Message
Backup Server
Lines
Representation
Normal Workflow
Fall-back Workflow

Figure 3.1: System Design using Local Infrastructure without Collaboration

3.1.2 Distributed Cyberbullying Detection using Local Infrastructure with Collaboration

Fig. 3.2 shows the architecture which also uses local infrastructure. In this architecture, each bullying detection server uses a different ML algorithm to detect cyberbullying and returns results to the communication server. The communication server forwards an incoming message to all bullying detection servers in the system and combines the results from them based on a specific collaboration technique. In this approach, the bullying detection activity is spread across different bullying detection servers in parallel. The reason for implementing two different architectures have is to evaluate the performance of detection in a sequential and a parallel manner.

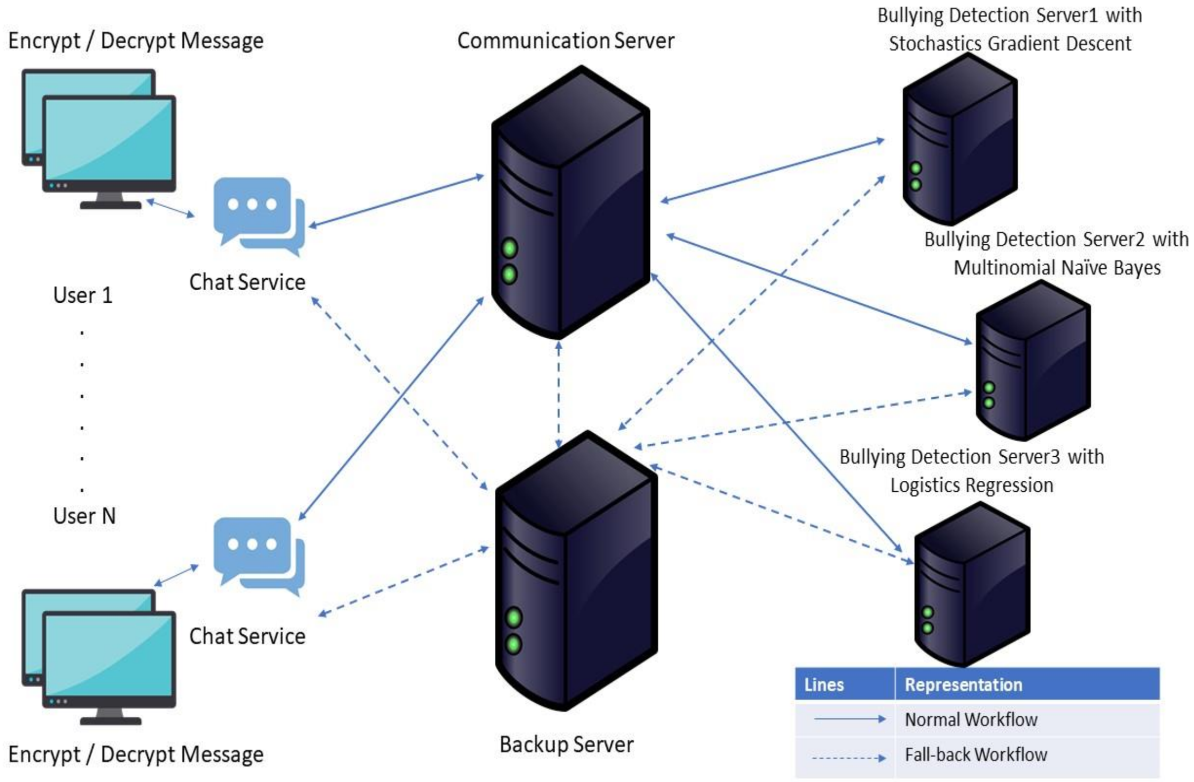

Encrypt / Decrypt Message
Communication Server
Bullying Detection Server1 with Stochastics Gradient Descent
Chat Service
User 1
Bullying Detection Server2 with Multinomial Naïve Bayes
User N
Bullying Detection Server3 with Logistics Regression
Chat Service
Encrypt / Decrypt Message
Backup Server
Lines
Representation
Normal Workflow
Fall-back Workflow

Figure 3.2: System Design using Local Infrastructure with Collaboration

3.1.3. Distributed Cyberbullying Detection using AWS Elastic Beanstalk without Collaboration

Fig. 3.3 shows the architecture which uses the AWS Elastic Beanstalk cloud service [15]. In this architecture, each bullying detection server uses all the ML algorithms to detect cyberbullying and returns a combined result to the communication server. Again, the bullying detection server executes ML algorithms in a sequential manner. In this architecture, the communication server forwards the message to a load balancer and, based on the current load, the message gets transferred to an appropriate bullying detection server. We have decided to use the cloud infrastructure due to its elasticity and other advantages such as auto-scaling, low cost, high availability, and large computation power [36].

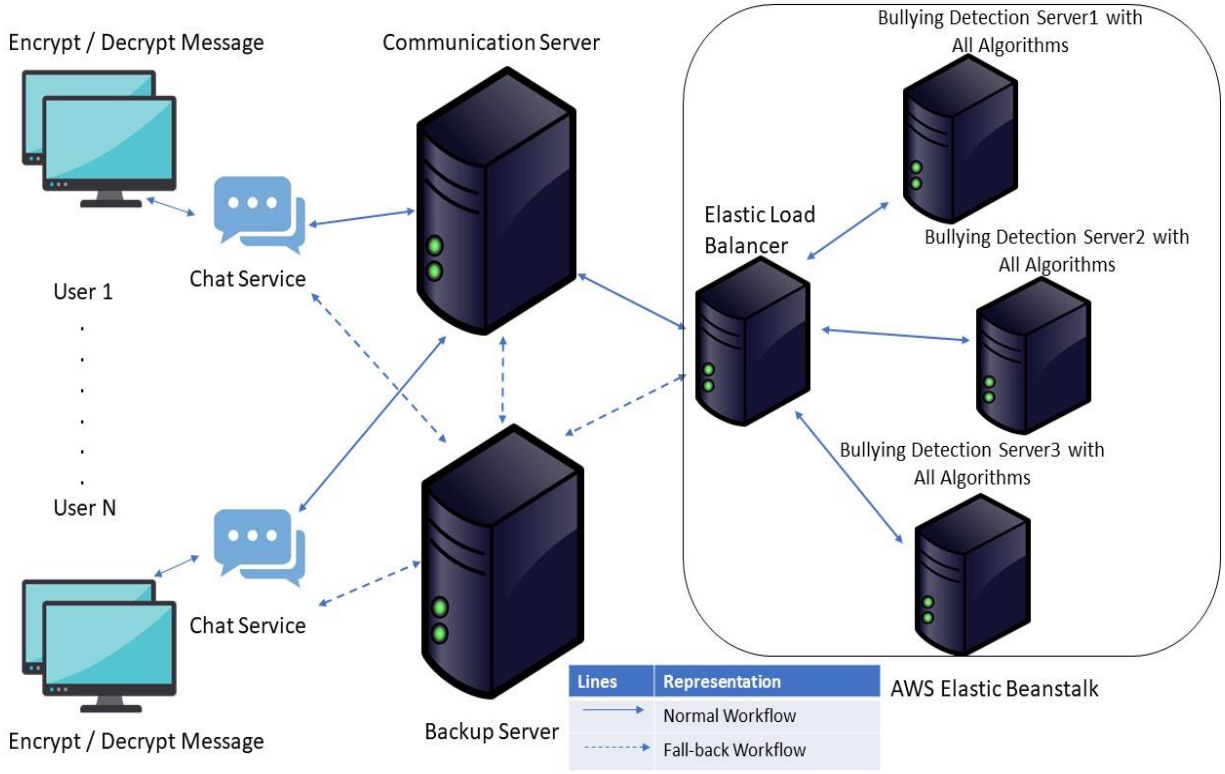

Encrypt / Decrypt Message
Communication Server
Bullying Detection Server1 with All Algorithms
Chat Service
Elastic Load Balancer
Bullying Detection Server2 with All Algorithms
User 1
.
.
.
.
.
.
User N
Bullying Detection Server3 with All Algorithms
Chat Service
Encrypt / Decrypt Message
Backup Server
Lines
Representation
Normal Workflow
Fall-back Workflow
AWS Elastic Beanstalk

Figure 3.3: System Design using AWS Elastic Beanstalk without Collaboration

3.1.4 Distributed Cyberbullying Detection using AWS Elastic Beanstalk with Collaboration

Fig. 3.4 shows the architecture which again uses the AWS Elastic Beanstalk cloud service [15]. However, in this architecture, each bullying detection server uses a different ML algorithm to detect cyberbullying and returns its result to the communication server. The communication server forwards an incoming message to all the bullying detection servers via the load balancer and combines results from them based on a specific collaboration technique. In this approach, the bullying detection activity is spread across different bullying detection servers in parallel. Again, the reason for these two different architectures is to estimate the performance of recognition in a serial manner and in a parallel manner.

We describe various experiments, performed using these architectures, and their analyses in the next chapter.

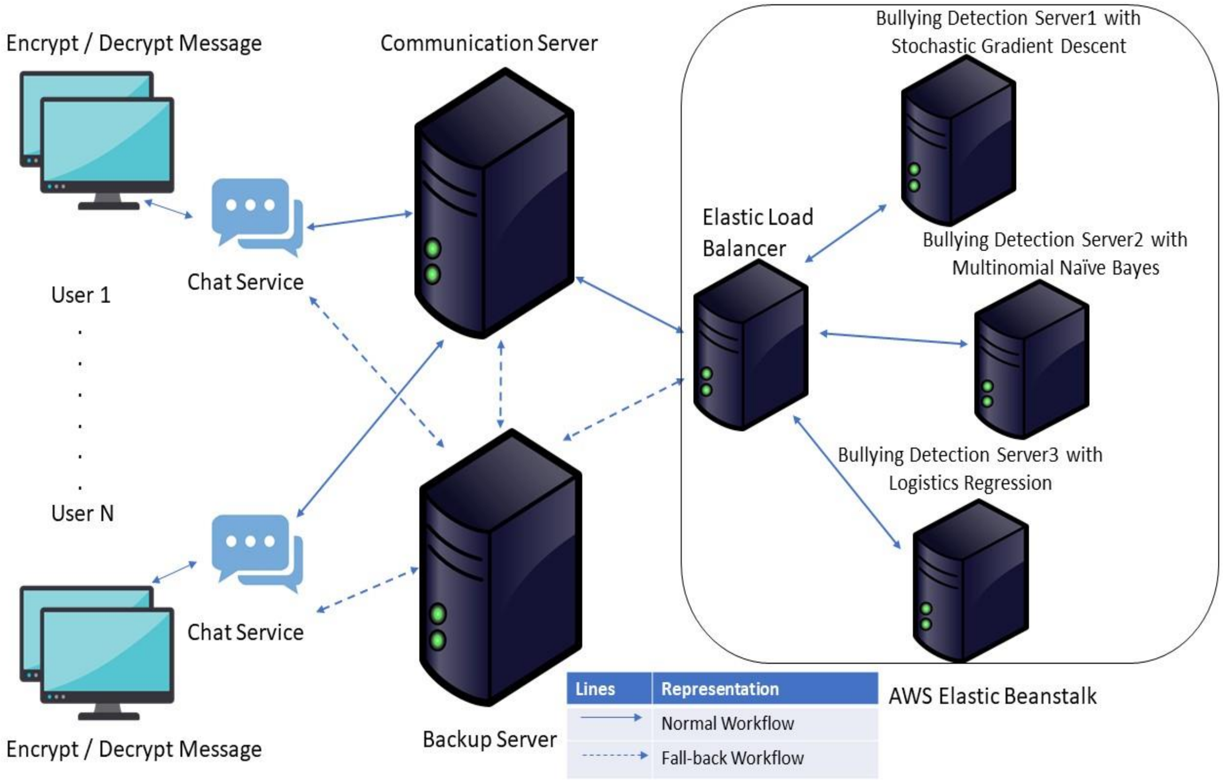

Encrypt / Decrypt Message
Communication Server
Bullying Detection Server1 with Stochastic Gradient Descent
Elastic Load Balancer
Bullying Detection Server2 with Multinomial Naïve Bayes
Chat Service
User 1
.
.
.
.
.
.
User N
Bullying Detection Server3 with Logistics Regression
Chat Service
AWS Elastic Beanstalk
Lines
Representation
Normal Workflow
Fall-back Workflow
Backup Server
Encrypt / Decrypt Message

Figure 3.4: System Design using AWS Elastic Beanstalk with Collaboration

3.2 System Workflow

Fig 3.5 shows the overall system workflow. We start with the data gathering phase – we have gathered data from multiple sources such as newspaper reviews, tour reviews, and tweets from multiple languages. The next step in the process is to use preprocessing techniques and remove stop-words and un-necessary characters. Also, we convert all text into lower case, so that the same words with different cases are treated identically. Once we have acquired data, we train the ML model using the preprocessed data. We have used either 3 ML algorithms (Multinomial Naïve Bayes (MNB), Stochastics Gradient Descent (SGD), and Logistic Regression (LR)) as shown in Fig 3.5 – details of the algorithms are provided in Chapter 4. In the next step, we have used 4 different architectures (described in the previous section) to test our proposed approach – each architecture results in a binary classification (bullying or non-bullying). For the two collaborative architectures, we perform the merging of the results.

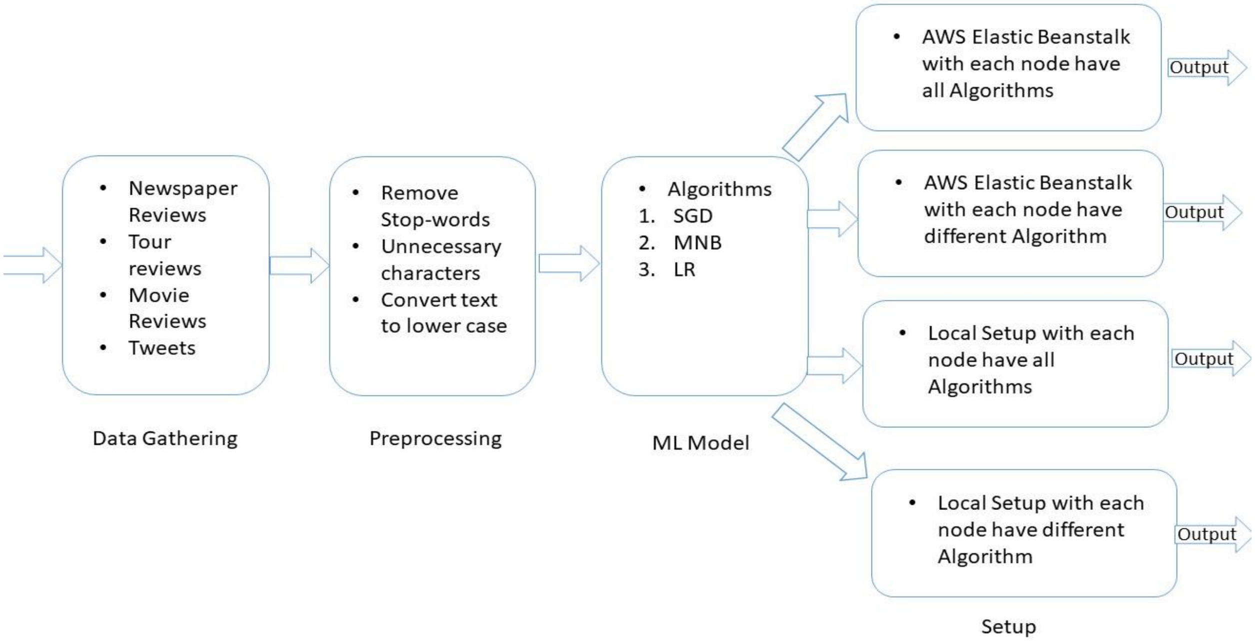

Newspaper Reviews
Tour reviews
Movie Reviews
Tweets
Data Gathering
Remove Stop-words
Unnecessary characters
Convert text to lower case
Preprocessing
Algorithms
1. SGD
2. MNB
3. LR
ML Model
AWS Elastic Beanstalk with each node have all Algorithms
AWS Elastic Beanstalk with each node have different Algorithm
Local Setup with each node have all Algorithms
Local Setup with each node have different Algorithm
Output
Output
Output
Output
Setup

Figure 3.5: System Workflow Diagram

3.3 Performance Metrics

We are using accuracy and F1-score [37] (similar to Mangaonkar et al. in [32]), performance metrics to inspect and analyze the performance of various ML- and NLP-based classification techniques. These metrics utilize True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Their computational formulae are indicated below [37] – “T” in these formulae indicated the summation of TP, FP, TN, and FN. In addition, we are using end-to-end time as another metric to analyze and compare the time taken by each algorithm.

1. Accuracy: Accuracy measures the amount of accurate predictions made by the model. It is formulated as:

   $$\text{Accuracy} = (TP + TN) / T \quad (1)$$

2. Precision: Precision is the measure of bullying tweets correctly predicted by the algorithm. It is formulated as:

   $$\text{Precision} = TP / (TP + FP) \quad (2)$$

3. Recall: Recall is the ratio of how many bullying tweets, out of all available ones, are actually detected by the algorithm. It is formulated as:

   $$\text{Recall} = TP / (TP+FN) \quad (3)$$

4. F1-Score: F-score gives an unbiased class-wise result. It is calculated as:
   F1 = 2*((Precision * Recall) / (Precision + Recall)) (4)
5. Time: This metric computes the total time taken by an algorithm to classify the messages.

In a real-life scenario, the number of bullying messages is far less when compared to non-bullying messages and hence, accuracy cannot be the correct metric when the dataset is imbalanced. For instance, if data contains 85% non-bullying data and if the model classifies all of messages as non-bullying, then as per equation (1) we still get an accuracy of 85%. Hence, we have used F1-Scores as a performance measure since it gives an unbiased class-wise result, which is important in our system. Although, typical chat applications are not very time critical, for the sake of completeness, we have carried out scalability experiments using the local infrastructure and the cloud infrastructure.

3.4 Data Collection

The first task in the multilingual cyberbullying detection is the collection of data. We have gathered data, as mentioned earlier, from multiple sources, which include tweets, newspaper reviews, and tourist reviews. We gathered data for three languages – English, Hindi and Marathi. In our previous work [35] for English language texts, we had downloaded data from the *formspring.me* [38] website – we used the same dataset for the English-related experiments. This dataset contains 40,900 messages out of which only 3,000 messages were bullying one. Also, this dataset was already tagged.

For the Hindi language-based dataset, we obtained data from different domains and on different topics. These include movie reviews [39], tour reviews [40], and newspaper reviews [41] on controversial topics such as harassment. The movie review dataset [39] has 245 reviews; the tour review dataset [40] has 192 reviews; and we manually obtained 184 newspaper reviews from [41] on harassment to create the combined dataset for our study. Hence, in all, for the Hindi-related study we gathered 621 reviews; tagged all these reviews manually. For the Marathi-related study, we again obtained data from multiple sources. The Marathi tour review dataset [42] has 106 records; we also collected newspaper reviews from multiple sources [43], which contain 196 reviews. Apart from these two data sources, we downloaded 508 tweets using Twitter’s API [44]. For the Marathi-related study, in all, we collected 810 reviews.

Indian languages, similar to other natural languages, are context sensitive and hence, to ensure correct labelling of all messages including sarcastic messages, we manually labelled all the messages for both the Hindi and Marathi datasets. We added a field called “bullying” (i.e., output label) – if the value of this attribute is “1”, it means that the text is bullying in nature and a value of “0” means the text is non-bullying. This attribute is needed to train our ML model and to validate the performance of the model.

Our datasets for both languages contained approximately 9% bullying messages after tagging the data. This is an example of data imbalance and it can cause issues and we can achieve good accuracy just by predicting non-bullying class every-time, but this provides useless classifier for intended use case [45]. Hence, to overcome the data imbalance problem, we decided to generate additional instances of bullying messages from existing instances. Such an approach resulted in synthesized data sets. To generate these synthesized data sets, we performed the following steps:

- We stored the pre-processed cyberbullying messages into a list.
- We decided the number of additional instances to be incorporated into the datasets. We decided to double these instances so that the resulting dataset will have at least 20% bullying messages. We decided to generate at least 1/5th of bullying instances to train and test the model. However, there is no agreement about the incorporation of such additional data. In past work [35], we carried out experiments by generating 20% of such additional bullying data and obtained good results. Hence, we followed the same technique in this thesis also.

Bag of Words (BoW), as indicated in the second chapter, is required for a lexicon-based approach. Hence, we have gathered *senti-wordnet* for all languages. We downloaded *senti-wordnet* from [46] for English, and *hindi-wordnet* from [47] for Hindi. Wordnet contains a list of words, sentiment and weights associated with it. For Marathi we have created *senti-wordnet* on our own, as there is no accepted word list publically available. We translated *hindi-wordnet* [47] into *marathi-wordnet* using Google translation API [48]. In addition, we obtained lists of positive and negative words in Marathi from [49]. Also, we obtained a list of abusive words and swear words from [50] and assigned weights to these words by taking the average of negative sentiment words. This process has resulted in the creation of *marathi-wordnet* for our study.

For our study involving ML algorithms, we have used 80% of data to train the model and 20% data for testing the model in all of the languages. For the lexicon-based method, we have used the BoW and 100% of the data for testing.

3.5 Model Creation with Scikit-Learn Algorithms

Scikit-learn machine-learning algorithms [51] are used in this study. Different classification algorithms are supported by the Scikit-learn machine learning algorithms such as Multinomial Naive Bayes, Stochastic Gradient Descent, and Logistics Regression. These are supervised machine learning algorithms. These algorithms were selected for this thesis, as they perform well on the Topic Modeling and Text Classification tasks as specified in our past work [35] as well as in literature (refer chapter 2). The models for classification of cyberbullying tweets we developed by training machine learning algorithms. We trained and tested the machine-learning model on datasets that were explained in the previous section.

The following is the process that we employed while creating the machine-learning model:

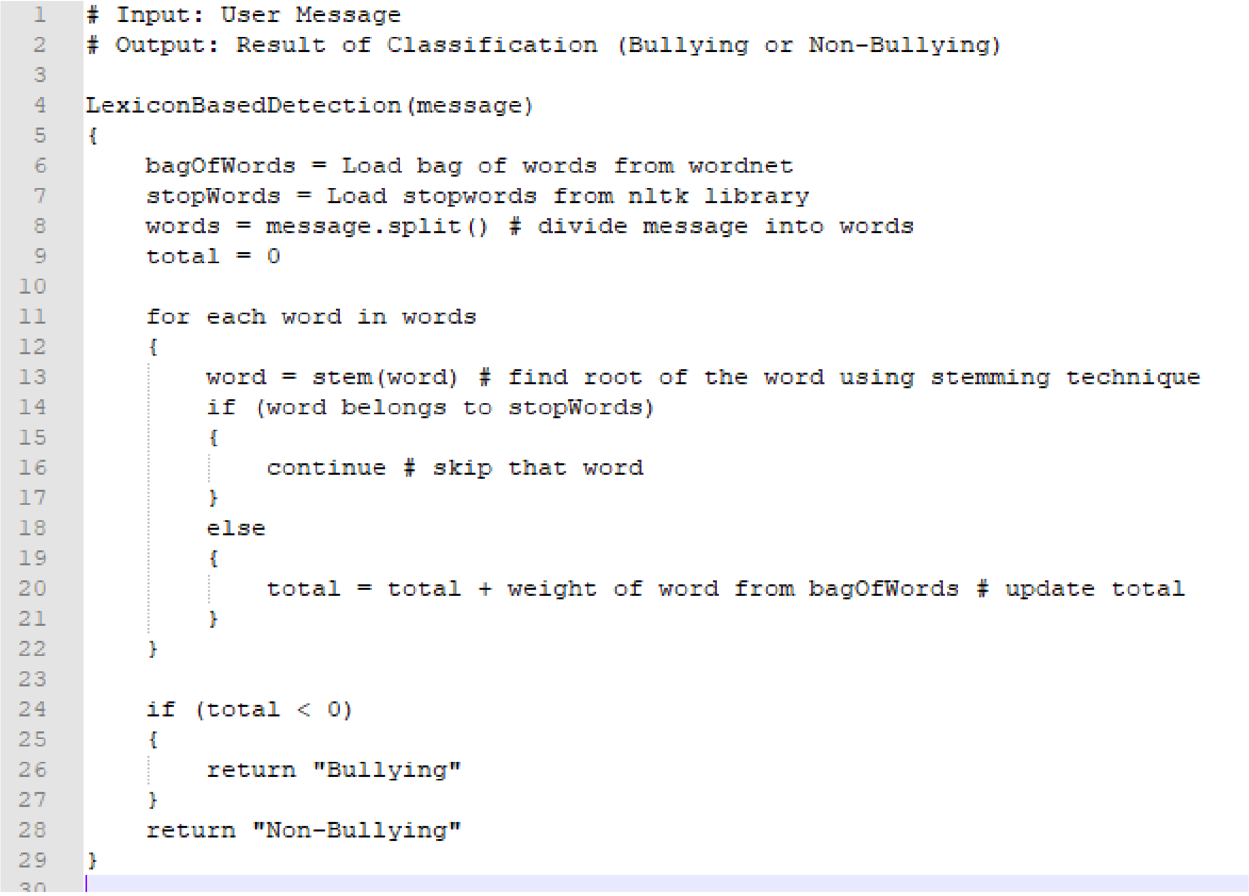
1. Input dataset: We have obtained data from different sources as discussed in Section Data Collection.
2. Preprocessing: We could not use data obtained from different sources in their native form due to various reasons such as the presence of stop words and special characters. Hence, we removed these stop words (e.g., a, and, the) and unnecessary characters (e.g., #, @ and URLs).
3. Train the model: We then divide the dataset and use 80% of the data set for training purposes. We have performed 10-fold Cross Validation for all our experiments. This basically means that each data point appears 9 times in train data and exactly once in test data. This is done so that, no matter how the data is divided, we always compute the average error across the folds to get a generalized score [52].
4. Test: Finally, we predict the outcome (cyberbullying or non-cyberbullying) on the remaining 20% of the data set using the trained model.

3.6 Model Creation with Lexicon-Based Algorithm

This method is based, as indicated earlier, on a simple Bag-of-Words technique. In this, a corpus of sensitive, abusive and hateful words is created. The algorithm (described below) uses this corpus to check for the presence of these words in messages to detect bullying. We have obtained *sentiword-net* from [46] [47], added bullying words into it and assigned negative weights to it. The detailed algorithm to detect bullying is discussed in next section.

3.6.1 Pseudo-Algorithm for Lexicon Based Algorithm:

We are using the *LexiconBasedDetection()* method to classify message as bullying or non-bullying. This method takes a user message as input and returns the classification result. The pseudo-Algorithm of this method is as shown in Fig 3.6.



```
# Input: User Message
# Output: Result of Classification (Bullying or Non-Bullying)

LexiconBasedDetection(message)
{
    bagOfWords = Load bag of words from wordnet
    stopWords = Load stopwords from nltk library
    words = message.split() # divide message into words
    total = 0

    for each word in words
    {
        word = stem(word) # find root of the word using stemming technique
        if (word belongs to stopWords)
        {
            continue # skip that word
        }
        else
        {
            total = total + weight of word from bagOfWords # update total
        }
    }

    if (total < 0)
    {
        return "Bullying"
    }
    return "Non-Bullying"
}
```

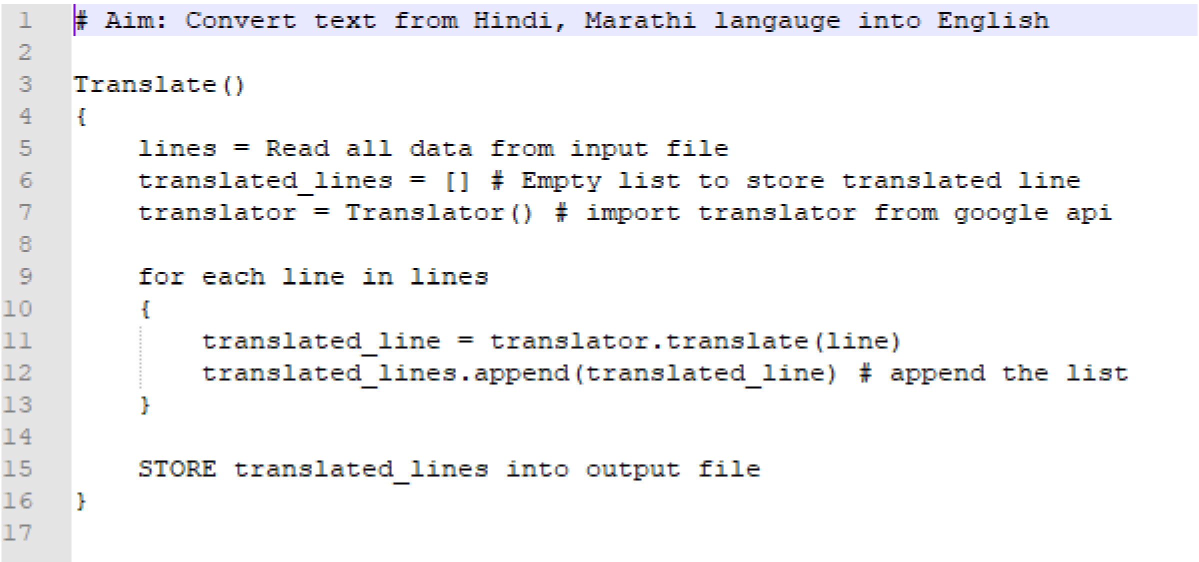
Figure 3.6: Pseudo-Algorithm for Lexicon-based Detection

3.7 Model Creation using Translator

This method is based on use of a translator. We have used Google translator API [48] to translate text in other languages to English and use techniques discussed in section 3.6 and 3.7 post translation. The purpose of this work is to generate a generic model which is built based on English input. Hence, there is no need to maintain datasets of different languages and train models for other languages. The detailed algorithm to translate text into English is discussed in next section.

3.7.1 Pseudo-Algorithm for Translating Text to English:

We use the *Translate ()* method to translate messages to English. This method reads a user message from the input file and stores the translated text back to the output file. This method uses the Google's Translator API [48]. The pseudo-Algorithm of this technique is as shown in Fig 3.7.



```
# Aim: Convert text from Hindi, Marathi langauge into English

Translate()
{
    lines = Read all data from input file
    translated_lines = [] # Empty list to store translated line
    translator = Translator() # import translator from google api

    for each line in lines
    {
        translated_line = translator.translate(line)
        translated_lines.append(translated_line) # append the list
    }

    STORE translated_lines into output file
}
```

Figure 3.7: Pseudo-Algorithm to Translate Text to English

3.8 Parameters in Distributed-Collaborative Detection Approach

This section explains various constraints involved in the collaborative detection of cyberbullying, which impact the performance of the system. Detailed experiments with variation in these parameters are discussed in next chapter.

3.8.1 Nodes in the Network

The count of nodes in the network which participate in the collective detection of cyberbullying behavior. In our system, the message server forwards the request to bullying detection node.

3.8.2 Training Set Associated with Each Node

In our system, all nodes participating in collaboration use the same dataset for training. We have chosen to use same dataset to train each ML model to simplify the design. Mangaonkar et al. in [32] carried out experiments with unified and different training datasets. All such variations are beyond the scope of this thesis, as the main focus of the thesis is the multi-language detection of cyberbullying behavior.

3.8.3 Number of Opinions

The communication server seeks opinions from different bullying detection nodes to determine cyberbullying. In our experiments, we have used three ML algorithms in these nodes and the communication server takes an opinion from all of them.

3.8.4 When to Collaborate

Since the communication server takes opinions from all nodes, all the nodes participate in collaboration to classify each message to advance the total accuracy of the cyberbullying detection.

3.8.5 Result Merging Technique

The result merging techniques are important to join results of different nodes, because they help to improve the accuracy of the prediction. We have used three opinion merging techniques in this research.

1) OR Merging: If any detection node categorizes a tweet as bullying, the communication server considers that message as bullying and warns the sender.
2) AND Merging: If all detection nodes classify a tweet as bullying, then only the communication server considers that message as bullying and warns the sender.
3) Majority Voting: In this case, a message is classified as bullying only when more than N/2 nodes detect a message as bullying, where the number of nodes is N participating in

collaboration (in our case it is 3). Based on the majority, the communication server decides to warn the sender.

The OR merging technique increases the false positives; the AND merging technique increases false negatives; and the poor performing nodes in the system are covered by the majority voting by overriding their classification. The accuracy of the classification differs based on the merging technique. Thus, it is an important parameter to select in collaborative detection of cyberbullying.

3.9 Limitations of our system

Our approach has the following limitations:

- As indicated earlier, sarcasm detection is out of the scope of our proposed system.
- The system can handle messages from only one language at a time.
- There is no publically available ground truth associated with the datasets and their tagging. However, to mitigate our bias to a small extent, we asked another native speaker to re-tag the messages and we validated their tagging against our tagging.

The proposed architectures, associated algorithms and the datasets discussed in this chapter are used for the experimental work which is described in the next chapter.

CHAPTER 4. EXPERIMENTAL RESULTS

This chapter comprises of multiple experiments that were carried out in order to empirically authenticate the proposed multilingual cyberbullying detection method to do cyberbullying detection in online social media networks such as Twitter.

4.1 Multilingual Cyberbullying Detection Approach

In this section, we discussed results obtained from different machine learning algorithms, a lexicon-based algorithm and from using translator on three languages – English, Hindi, and Marathi.

4.1.1 Experimental work with English Text

4.1.1.1 Machine Learning Techniques

We ran 4 ML algorithms, Multinomial Naïve Bayes (MNB), Support Vector Machine (SVM), Stochastics Gradient Descent (SGD), and Logistic Regression (LR), (refer to Section 3.3) on the dataset obtained from Form-spring [36]. This dataset contains 40,900 messages, out of which 3000 messages are bullying ones. Since, very few bullying instances are present in the dataset, we decided to generate additional instances of bullying data using the data synthesis technique. We created additional 17,000 instances so that the synthesized data will contain 1/3rd bullying instances – this avoids the data imbalance issue.

Table 4.1 indicates the results of our experiments. These results show that LR outperforms other algorithms and achieves an accuracy of 94%. In our previous work, we have used SGD and MNB algorithms on the same dataset with synthesized data and had achieved an accuracy of 93% and 88% respectively [1]. In addition, as seen from Table 4.1, the accuracy of the ML model (for all the four algorithms) increases when the size of the dataset is increased. The reason for this behavior is that with the synthesized data, we are able to train the model on additional items.

Table 4.1. Performance of the ML Algorithms on the Formspring Dataset

No	Algorithm	Synthesize Data	Accuracy	Precision	Recall	F1-Score
1	Multinomial Naïve Bayes (MNB)	No	0.8780	0.8865	0.8780	0.8792
		Yes	0.8845	0.8974	0.8895	0.8845
2	Support Vector Machine (SVM)	No	0.9185	0.9215	0.9186	0.9173
		Yes	0.9275	0.9365	0.9236	0.9259
3	Stochastics Gradient Descent (SGD)	No	0.9232	0.9257	0.9045	0.9177
		Yes	0.9352	0.9365	0.9135	0.9263
4	Logistic Regression (LR)	No	0.9311	0.9311	0.9312	0.9307
		Yes	0.9424	0.9421	0.9438	0.9412

We carried out another experiment using the ML model on the Formspring dataset. In this experiment, we did not generate additional bullying instances; instead, we reduced the sample set of the non-bullying instances. We decided to keep 9,000 non-bullying instances and 3,000 bullying ones. The outcomes of the study are as shown in Table 4.2. The results show that the LR algorithm, again, outperforms all other algorithms. However, if we compare the results of Table 4.1 and 4.2, there is slight decrease in the accuracy of the results when the dataset is reduced. Hence, we can conclude that providing more data to train the ML models can help to improve the accuracy.

Table 4.2. Performance of the ML Algorithms on a Subset of the Formspring Dataset

No.	Algorithm	Accuracy	Precision	Recall	F1-Score
1	MNB	0.8774	0.8758	0.8774	0.8762
2	SVM	0.8834	0.8864	0.8823	0.8895
3	SGD	0.8937	0.8926	0.8938	0.8920
4	LR	0.9024	0.9047	0.9075	0.9056

4.1.1.2 Lexicon-based Technique

For this study, we used the same Formspring dataset. The results of the lexicon-based experiments are shown in Table 4.3. As seen from Tables 4.1 and 4.3, the ML techniques outperform the lexicon-based approach. There is a large difference (around 13%) between the results shown in Table 4.1 and the ones described in Table 4.3. One possible reason for this difference could be that the lexicon-based technique completely depends on the dictionary of words and the associated weights. Another possible reason could be that people use slang language/mis-spelled words while texting and those words might not appear in the dictionary. Hence, chances are high that the total bullying score of a text message may not reflect the correct sentiment implied by the sender.

Table 4.3. Results of Lexicon Based Algorithm on the Formspring Dataset

No.	Algorithm	Accuracy	Precision	Recall	F1-Score
1	Lexicon Based	0.8106	0.8376	0.8107	0.8237

4.1.1.3 Collaboration Techniques

For this study, we have used the Formspring dataset and 3 ML algorithms i.e., SGD, MNB, LR. In this experiment, we have chosen the two top performing and one poor performing algorithm. Also, we arbitrarily decided to drop the SVM algorithm to reach consensus in the majority voting collaboration scheme. All these trials are carried out on the synthesized data and we used the 3 collaboration techniques discussed in Section 3.9. The results of collaboration are as shown in Table 4.4. These results show that the AND collaboration produces more false negatives because if any node wrongly categorizes a tweet as non-bullying, then the overall result becomes non-bullying. The OR collaboration technique produces more false positives, since it classifies a tweet as abusive if even a single node labels that tweet as bullying. In majority voting, it classifies a tweet as bullying or non-bullying based on the majority i.e., if 2 out 3 nodes classify a tweet as bullying then the overall result is classified as bullying and vice-versa. The Majority voting collaboration technique balances out the classification by covering the outcomes of a poorly performing node. The results of our experiments show the exact same behavior, which is as expected. If we compare the results of collaboration with stand-alone ML algorithms (refer Tables 4.1 and 4.4), the LR algorithm still outperforms all collaboration techniques. However, if we look at other algorithms, they produce results that are close to the results of the collaboration technique. This means collaboration techniques can help to advance the overall performance of the system even if the system contains a few poor-performing nodes.

Table 4.4. Performance of Collaboration Techniques on Formspring Dataset

No.	Algorithm	Accuracy	Precision	Recall	F1-Score
1	AND	0.9127	0.9188	0.9127	0.9108
2	OR	0.8927	0.8988	0.8927	0.8908
3	Majority	0.9307	0.9317	0.9308	0.9302

4.1.2 Experimental work with Hindi Text

4.1.2.1 Machine Learning Techniques

As specified in the previous chapter, the dataset for these experiments was set up by collecting messages from different domains and different topics. These includes movie reviews [37], tour reviews [38], and newspaper reviews [39] on controversial topics such as harassment. The movie review [37] dataset has 245 reviews, the tour review [38] dataset has 192 reviews and we manually obtained 184 newspaper reviews from [39] on harassment. After manually tagging original data, it was noted that the dataset contained only 9% bullying instances. Hence, we decide to double these instances using the earlier mentioned data synthesis technique to avoid the data imbalance issue. We carried out experiments on these datasets, again using the same 4 ML algorithms. The outcomes of the experiments are as exposed in Table 4.5. These outcomes show that the LR algorithm, similar to the English-based experiments, outperforms all other algorithms. However, the resulting numbers are actually less than the English-related numbers because the Hindi dataset is very small. In addition, synthesized data improves the performance of the ML models for all 4 algorithms.

4.1.2.2 Lexicon-based technique

For this experiment, we have used the same Hindi dataset. The results of the lexicon-based technique are shown in Table 4.6. From this table, the ML techniques outperform the lexicon-based approach. There is a significant difference (around 40% in general) between the results shown in Table 4.5 and the results indicated in Table 4.6. In addition, if we compare the results of a lexicon-based approach for English and Hindi (refer to Table 4.3 and Table 4.6), we can notice a large difference (around 30%). The reason for this behavior is that there are very few publically available resources about Hindi language and to the finest of our knowledge, there are no previous attempts made to detect cyberbullying behavior in Hindi (refer chapter 2). For example, the Hindi

Table 4.5. Results of ML Algorithms on the Hindi Dataset

No	Dataset	Algorithm	Synthesize Data	Accuracy	Precision	Recall	F1-Score
1	Movie Reviews	SGD	No	0.7346	0.7502	0.7347	0.7347
		MNB	No	0.6734	0.6735	0.6735	0.6735
		SVM	No	0.7046	0.7126	0.7023	0.7034
		LR	No	0.7346	0.7346	0.7346	0.6933
		SGD	Yes	0.7391	0.7801	0.7391	0.7441
		MNB	Yes	0.7681	0.7636	0.7681	0.7631
		SVM	Yes	0.7423	0.7476	0.7498	0.7422
		LR	Yes	0.7826	0.7826	0.7826	0.7626
2	Tour Reviews	SGD	No	0.7948	0.7985	0.7949	0.7946
		MNB	No	0.7717	0.7729	0.7718	0.7718
		SVM	No	0.7835	0.7815	0.7864	0.7892
		LR	No	0.7979	0.7923	0.7987	0.7934
		SGD	Yes	0.9322	0.9322	0.9322	0.9322
		MNB	Yes	0.9491	0.9527	0.9492	0.9479
		SVM	Yes	0.9302	0.9386	0.9387	0.9368
		LR	Yes	0.9452	0.9435	0.9425	0.9415
3	Newspaper Reviews	SGD	No	0.4594	0.4669	0.4595	0.4618
		MNB	No	0.3513	0.3563	0.3585	0.3523
		SVM	No	0.4163	0.4173	0.4183	0.4196
		LR	No	0.5135	0.5285	0.5135	0.5149
		SGD	Yes	0.7719	0.7770	0.7719	0.7742
		MNB	Yes	0.8070	0.8050	0.8060	0.7970
		SVM	Yes	0.7936	0.7924	0.7942	0.7923
		LR	Yes	0.9122	0.9126	0.9123	0.9089

senti word net contains around 8,000 words, whereas the English sentiwordnet contains around 51,000 words.

Table 4.6. Results of Lexicon-based Algorithm on the Hindi Dataset

No	Dataset	Accuracy	Precision	Recall	F1-Score
1	Movie Reviews	0.4907	0.4904	0.4909	0.4905
2	Tour Reviews	0.5007	0.5004	0.5009	0.5005
3	Newspaper Reviews	0.4808	0.4804	0.4809	0.4805

4.1.2.3 Using Translator

In order to over the issue of scarcity of resources in Hindi, we decided to translate the Hindi text into English using Google's translator API [46]. For this experiment, we have used the same Hindi dataset. We converted this dataset using the translator and then ran the lexicon-based algorithm, provides the English *sentiwordnet* as an input to classify the translated dataset. The values of our experiment are as shown in Table 4.7. As seen from Table 4.7, translation does not improve the accuracy and other metrics associated with the cyberbullying detection. In addition, if we compare results from Tables 4.6 and 4.7, we only see a slight improvement in accuracy. One possible reason for the poor performance could be that the English *sentiwordnet* has almost 7 times more words in it than the Hindi *sentiwordnet*. Another reason for the poor performance is the inherent limitations of the translator. We noticed, during our experiments, that the translator failed many times to translate Hindi words and sometimes the meaning or the gist of the whole message was lost after translation. Hence, the use of translator did not yield better results.

Table 4.7. Results of Lexicon Based Algorithm on Translated Hindi Dataset

No	Dataset	Accuracy	Precision	Recall	F1-Score
1	Movie Reviews	0.5576	0.5532	0.5548	0.5556
2	Tour Reviews	0.5549	0.5567	0.5598	0.5569
3	Newspaper Reviews	0.5412	0.5486	0.5456	0.5434

4.1.2.4 Collaboration Techniques

For this study, we have used the same Hindi dataset. Again, we have used 3 ML algorithms, i.e., SGD, MNB, and LR. All the trials are carried out on synthesized data. We have used 3 collaboration techniques, i.e., AND, OR, and Majority voting discussed in Section 3.9. The results of the collaboration algorithms are shown in Table 4.8. For the Hindi dataset, the majority voting technique outperforms all other collaboration techniques as expected. This is because we have only one poor performer node (i.e., the MNB algorithm) in the system. The LR algorithm outperforms all other collaboration techniques for the Hindi datasets as well.

Table 4.8. Collaboration Techniques Performance on the Hindi Dataset

No.	Dataset	Algorithm	Accuracy	Precision	Recall	F1-Score
1	Movie Reviews	AND	0.7391	0.7933	0.7391	0.7438
		OR	0.7018	0.7654	0.7028	0.7143
		Majority	0.7971	0.7962	0.7971	0.7897
2	Tour Reviews	AND	0.9322	0.9362	0.9322	0.9330
		OR	0.9212	0.9255	0.9266	0.9243
		Majority	0.9322	0.9326	0.9322	0.9311
3	Newspaper Reviews	AND	0.7719	0.8548	0.7719	0.7853
		OR	0.7591	0.8256	0.7335	0.7597
		Majority	0.7894	0.7895	0.7895	0.7895

4.1.3 Experimental work with Marathi Text

4.1.3.1 Machine Learning Techniques

For this experiment, as indicated earlier, we have obtained datasets from multiple sources – the Marathi tour review [40] has 106 records, and we collected newspaper reviews from multiple sources [41], which has 196 reviews. Apart from these two sources, we have downloaded 508 tweets using the Twitter’s API [42]. After manually tagging the original data, we found that the dataset contains only 9% bullying instances. Therefore, we decide to double these instances using the data synthesis technique to avoid the data imbalance issue (similar to the Hindi and English datasets). We carried out experiments on these datasets using the same 4 ML algorithms, as in the previous two situations. The outcomes of these experiments are as shown in Table 4.9. Again, similar to the Hindi and English datasets, the results in Table 4.9 show that the LR algorithm

Table 4.9. Results of ML Algorithms on the Marathi Dataset

No	**Dataset**	**Algorithm**	**Synthesize Data**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
1	Tour Reviews	SGD	No	0.9523	0.9549	0.9524	0.9483
		MNB	No	0.9523	0.9643	0.9524	0.9551
		SVM	No	0.9148	0.9175	0.9158	0.9176
		LR	No	0.9571	0.9524	0.9563	0.9588
		SGD	Yes	0.9124	0.9219	0.9024	0.9092
		MNB	Yes	0.9012	0.9052	0.9012	0.9046
		SVM	Yes	0.9053	0.9073	0.9054	0.9027
		LR	Yes	0.9756	0.9235	0.9574	0.9575
2	Twitter Tweets	SGD	No	0.8157	0.9433	0.8158	0.8749
		MNB	No	0.7894	0.9423	0.7895	0.8591
		SVM	No	0.7944	0.7924	0.7975	0.7987
		LR	No	0.8236	0.9323	0.8236	0.8954
		SGD	Yes	0.9482	0.9536	0.9483	0.9484
		MNB	Yes	0.9455	0.9480	0.9455	0.9456
		SVM	Yes	0.9447	0.9486	0.9472	0.9489
		LR	Yes	0.9655	0.9648	0.9668	0.9688
3	Newspaper Reviews	SGD	No	0.7037	0.7037	0.7037	0.7037
		MNB	No	0.7077	0.7056	0.7078	0.7054
		SVM	No	0.7196	0.7154	0.7185	0.7165
		LR	No	0.8518	0.8186	0.8578	0.8234
		SGD	Yes	0.9148	0.9172	0.9149	0.9143
		MNB	Yes	0.9161	0.9167	0.9162	0.9160
		SVM	Yes	0.9146	0.9165	0.9149	0.9186
		LR	Yes	0.9574	0.9598	0.9527	0.9572

outperforms other algorithms. In addition, again similar to the past two situations, synthesizing data improves the performance of ML models (as shown in Table 4.9).

4.1.3.2 Lexicon-based technique

For this experiment, we have used the same Marathi dataset as used for the ML techniques. In addition, we translated Hindi wordnet [45] into Marathi wordnet using Google's translation API [46]. We also obtained a list of positive and negative words in Marathi from [47] and compiled another list of abusive and swear words from [48]; we have assigned weights to these words. These steps resulted in our own Marathi wordnet consisting of around 13,000 words, which are used for lexicon-based experiments.

The results of the lexicon-based experiment is as shown in Table 4.10. As seen from Tables 4.9 and 4.10, the ML techniques outperform the lexicon-based approach – this is similar to what we observed in the case of the Hindi experiments. Again, there is a large difference (around 20%) between the results shown in Tables 4.9 and 4.10. If we compare the results of the lexicon-based approach for English and Marathi (Table 4.3 and Table 4.10), we can see little difference (around 10%) between them. This is because, similar to the Hindi situation described earlier, there are limited publicly available resources for detecting cyberbullying behavior in Marathi language text and to the best of our knowledge, there is not even a single attempt reported in the literature that deals with Marathi text. If we compare the results of the experiments on the Hindi and Marathi datasets, using the lexicon-based approach (refer Table 4.6 and 4.10), the results for the Marathi dataset are better than the results obtained on the Hindi dataset. This is not surprising because we continuously added missing words to the Marathi wordnet while running experiments.

Table 4.10. Results of Lexicon Based Algorithm on Marathi Dataset

No	**Dataset**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
1	Tour Reviews	0.7238	0.7087	0.7238	0.6319
2	Twitter Tweets	0.7358	0.7038	0.7358	0.7156
3	Newspaper Reviews	0.7264	0.8963	0.7265	0.7916

4.1.3.3 Collaboration Techniques

For this study, we have used the same Marathi dataset. Also, we have used 3 ML algorithms i.e., SGD, MNB, and LR. All these experiments are carried out on synthesized data. We have used 3 collaboration techniques, i.e., AND, OR, and Majority voting discussed in Section 3.9. The results of the collaboration approach are shown in Table 4.11. For the Marathi dataset also, majority voting technique outperforms all other collaboration techniques as expected.

Table 4.11. Collaboration Techniques Performance on Marathi Dataset

No.	Dataset	Algorithm	Accuracy	Precision	Recall	F1-Score
1	Movie Reviews	AND	0.9656	0.9697	0.9656	0.9665
		OR	0.9566	0.9577	0.9596	0.9534
		Majority	0.9756	0.9797	0.9756	0.9765
2	Twitter Tweets	AND	0.9555	0.9575	0.9555	0.9554
		OR	0.9456	0.9492	0.9426	0.9422
		Majority	0.9627	0.9634	0.9628	0.9628
3	Newspaper Reviews	AND	0.9474	0.9411	0.9474	0.9476
		OR	0.9355	0.9313	0.9354	0.9376
		Majority	0.9574	0.9574	0.9574	0.9574

4.2 Performance Analysis

As indicated in Chapter 3, we have implemented 4 different architectures for the proposed detection system. Here we describe the performance analyses of these four options.

In the performance experiments, we have used 2 communication servers, 3 bullying detection servers, and 5 clients. We ran this experiment with the assumption that there are no failures in the system. In addition, in our experiment, the chat component is the same for all 4 architectures. Furthermore, we deployed 3 ML algorithms (i.e., MNB, SGD, and LR) on the bullying detection servers. The hardware configuration for the local distributed infrastructure was: CPU: Core i7, RAM: 8GB, Hard disk: 1TB; while for the AWS EBS cloud the configuration s: CPU: Intel AVX, RAM: 8 GB, Hard disk: 32 GB.

In the local/cloud with the sequential configuration, the 3 ML algorithms were deployed on each bullying detection server. Each server executes those three algorithms one after another, combines the results based on three collaboration techniques and sends the combined results back to the communication server. Whereas, in local/cloud with the parallel configuration only one ML algorithm is deployed on each bullying detection server. In this case, the communication server forwards the same message to all 3 bullying detection servers in parallel and combines their results based on the 3 collaboration techniques. The outcomes of our experiments are as shown in Table 4.12. In these experiments, we have used 20% of the dataset for testing for all the languages and calculated the average response time. In the case of local/cloud server with sequential configuration, we have observed that the bullying detection server requires twice the time to execute than the parallel configuration. This is because, in the parallel configuration, the communication server forwards the message to all bullying detection servers in parallel and combines the results. Communication time is little bit low for the local setup than the cloud setup, because in the local setup the nodes are in the same LAN; whereas, in case of cloud, nodes are deployed in the North Virginia zone (which is 500 miles away from the client). However, the cloud-based configurations due to the elastic nature of AWS and perhaps more powerful machines, perform the detection part faster than the local setup; thereby, achieving less end-to-end or total time.

Table 4.12 Performance Analysis of 4 Architecture in milliseconds

Operation	**Local Server with All Algorithms**	**Local Server with Only One Algorithm**	**Cloud Server with All Algorithms**	**Cloud Server with Only One Algorithm**
Communication	4.243	4.243	5.657	5.657
Bullying Detection	12.835	4.698	5.421	2.463
Total Time	17.078	8.941	11.078	8.12

4.3 Scalability Testing

In this experiment, we have used 2 communication servers, and 3 bullying detection servers with only one algorithm deployed on them for both local and cloud configurations. We decided to flood these two configurations with requests from multiple clients randomly. This experiment was carried out with 20, 40, 60, 80, and 100 clients (as shown in Figure 4.1). In Figure 4.1, the X-axis indicates the count of active clients and the Y-Axis serves as the average response time in milliseconds. Figure 4.1 shows that for both the configurations, the response time did not change significantly because the ML model is deployed on the Django REST framework [56]. This framework supports multiple client requests at a time and hence, allows the system to scale well [57]. Again, as expected, the cloud-based configuration performed better than the local configuration.

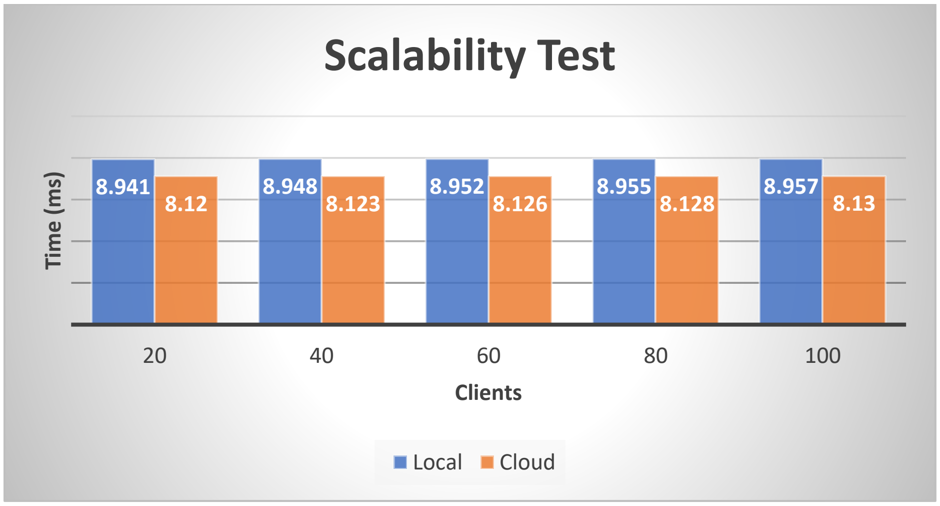


Figure 4.1 Scalability Test in milliseconds

4.4 Failure Testing

Failures are inherent in any distributed system. Hence, to measure the performance of our system in case of faults, we implemented three detection nodes and manually terminated one of them. Ten clients were created to concurrently send the requests to servers.

Table 4.13 Failure Testing in milliseconds

Operation	**Local Server with Failure**	**Cloud Server with Failure**	**Local Server without Failure**	**Cloud Server without Failure**
Communication	4.243	5.657	4.243	5.657
Bullying Detection	5.281	2.873	4.698	2.463
Total	9.524	8.53	8.941	8.120

Table 4.13 shows the performance comparison between with and without failure situations in local and cloud configurations. We have observed that the average response time increases when one server stops working because the other two servers must handle the same number of client requests as three servers (in the no failure scenario). Again, the cloud-based configuration performs better than the local configuration.

4.5 Ground Truth Validation of Dataset

As already mentioned in Chapter 3 (section Data Collection), we gathered data sets from multiple sources and we manually tagged them. As any such tagging is subject to a personal opinion and associated bias, hence, we invited another native speaker of Hindi and Marathi to tag the same datasets. We found that there was no difference in both the tagged versions except 4 messages in the Marathi newspaper review [43]. Hence, we decided to carry out an experiment on the Marathi newspaper review set [43] and see the impact of this different tagging on the results. For this experiment, we decided to use our best performing ML algorithm, i.e., LR. The outcomes of this experiment are as shown in Table 4.14. These results show that there is no significant difference due to tagging by different experts, since the number of tagging differences is very small.

Table 4.14 Validate Tagging

Tagged By	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Us	0.9574	0.9598	0.9527	0.9572
Native Speaker	0.9569	0.9695	0.9625	0.9570

4.6 Summary of Experiments

In the above sections we have discussed multilingual experiments with various configurations and compared their results. These results show that the ML technique outperforms the lexicon-based approach for all the languages. If we compare the results of the collaboration technique with ML techniques, the LR algorithm is the only algorithm that outperforms all collaboration techniques for all the languages. This means the collaboration techniques can help to advance overall detection performance of the system even if the system contains poor performing nodes. This observation is consistent with another study from the diabetes domain [58]. In addition, we observed that the cloud-based configurations always perform better than the local configurations.

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

This thesis has provided a multilingual (English, Hindi and Marathi) cyberbullying detection approach for detecting cyberbullying in text messages, tweets, tour and newspaper reviews. Also, this thesis concludes that the machine-learning approach, Logistic Regression, outperforms all other approaches in all the three languages. Also, the machine-learning approach with synthesized data outperforms in situations without the synthesized data. Various experiments with multiple server configurations have been carried out that indicate that the load balancer configuration works better than a single server configuration, and the cloud-based setups outperform the corresponding local setups. The following are the contributions of the thesis:

- A Multilingual Cyberbullying Detection approach was developed and empirically validated using experiments that were performed with English, Hindi, and Marathi datasets.
- The proposed Multilingual Cyberbullying Detection method was tested in the occurrence of failures, and various strategies and server configurations were implemented for collaborative and non-collaborative alternatives.
- It was observed that machine-learning techniques perform better than lexicon-based techniques across multiple languages while detecting the cyberbullying behavior. Also, it was noted that the performance of machine-learning techniques improves when combined with additional data generated using the data synthesize technique.
- It was reconfirmed that cloud-based detection outperforms the local setups for all experiments.
- Finally, it can be concluded that collaboration techniques can help improve overall performance of the system even if system contains poor nodes.

It is possible to advance this work in future endeavors. The future work includes testing the system on larger datasets and additional languages. In addition, combining multiple languages into a single sentence and see their effects on the accuracy of the models is another future work. Sarcasm detection, as indicated earlier, is a big challenge and its inclusion in the system will lead to another extension of this research. Incorporating a trust framework, to validate the data and eliminate any subjective bias, into the proposed system would also be a further enhancement.

REFERENCES

[1] Global digital report, 2018. Retrieved from: https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/. Last Accessed: 3/3/2019.

[2] What is Cyberbullying? Retrieved from: http://cyberbullying.org/what-is-cyberbullying. Last Accessed: 2/22/2019.

[3] P. Smith, J. Mahdavi, M. Carvalho, S. Fisher, S. Russell, and N. Tippett, “Cyberbullying: Its nature and impact in secondary school pupils,” Journal of child psychology and psychiatry, vol. 49, no. 4, pp. 376–385, 2008.

[4] S. Hinduja and J. W. Patchin, “Cyberbullying: Neither an epidemic nor a rarity,” European Journal of Developmental Psychology, vol. 9, no. 5, pp. 539–543, 2012.

[5] The Annual Bullying Survey, 2017. Retrieved from: https://www.ditchthelabel.org/wp-content/uploads/2017/07/The-Annual-Bullying-Survey-2017-2.pdf. Last Accessed: 2/22/2019.

[6] Cyberbullying Research Center, 2016. Retrieved from: https://cyberbullying.org/2016-cyberbullying-data. Last Accessed: 2/22/2019.

[7] Cyberbullying Research Center, 2015. Retrieved from: https://cyberbullying.org/2015-data. Last Accessed: 2/22/2019.

[8] McAfee Survey, Teens and the Screen study: Exploring Online Privacy, Social Networking and Cyberbullying, 2014. Retrieved from: https://www.marketwatch.com/press-release/cyberbullying-triples-according-to-new-mcafee-2014-teens-and-the-screen-study-2014-06-03. Last Accessed: 2/22/2019.

[9] Cox Internet Safety Survey, 2014. Retrieved from: https://www.cox.com/content/dam/cox/aboutus/documents/tween-internet-safety-survey.pdf. Last Accessed: 2/22/2019.

[10] Facebook Bullying Safety. Retrieved from: https://www.facebook.com/safety/bullying/. Last Accessed: 2/22/2019.

[11] Stopbullying.gov, USA goverment site. Retrieved from: https://www.stopbullying.gov/. Last Accessed: 2/22/2019.

[12] Cyber Bullying On The Rise: Facebook, Ask.Fm And Twitter The Most Likely Sources. (2015). Retrieved from: 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. Last Accessed: 2/22/19.

[13] Mobile phone users in countries; Wiki. Retrieved from: https://en.wikipedia.org/wiki/List_of_countries_by_number_of_mobile_phones_in_use. Last Accessed: 2/22/2019.

[14] Language Native Speakers Wiki. Retrieved from: https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers. Last Accessed: 2/22/2019.

[15] AWS Elastic Beanstalk. Retrieved from: https://aws.amazon.com/elasticbeanstalk/?sc_channel=PS&sc_campaign=acquisition_US&sc_publisher=google&sc_medium=ACQ-P%7CPS-GO%7CBrand%7CDesktop%7CSU%7CMachine%20Learning%7CElastic%20Beanstalk%7CUS%7CEN%7CText&sc_content=elastic_beanstalk_e&sc_detail=amazon%20elastic%20beanstalk&sc_category=Machine%20Learning&sc_segment=293647516945&sc_matchtype=e&sc_country=US&s_kwcid=AL!4422!3!293647516945!e!!g!!amazon%20elastic%20beanstalk&ef_id=EAIaIQobChMI6976y_Pm4AIVB4TICh0PBAEHEAAYASAAEgLs7fD_BwE:G:s. Last Accessed: 3/3/2019.

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

[17] T. Bosse, and S. Stam, (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.

[18] Cyber Bullying Statistics, (2017). Retrieved from: https://www.guardchild.com/ https://www.guardchild.com/cyber-bullying-statistics/. Last Accessed: 2/22/2019.

[19] S. Algar (2019). 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/. Last Accessed: 2/22/2019.

[20] Y. Silva, C. Rich, J. Chon and L. M. Tsosie, “BullyBlocker: An app to identify cyberbullying in Facebook,” 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), San Francisco, CA, 2016, pp. 1401-1405. doi: 10.1109/ASONAM.2016.7752430

[21] G. Fahrnberger, D. Nayak, V. Martha and S. Ramaswamy, "SafeChat: A tool to shield children's communication from explicit messages," 2014 14th International Conference on Innovations for Community Services (I4CS), Reims, 2014, pp. 80-86. doi: 10.1109/I4CS.2014.6860557

[22] M. Rybnicek, R. Poisel and S. Tjoa, "Facebook Watchdog: A Research Agenda for Detecting Online Grooming and Bullying Activities," 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, 2013, pp. 2854-2859. doi: 10.1109/SMC.2013.487

[23] Rethink app. Retrieved from: http://www.rethinkwords.com/. Last Accessed: 2/22/2019

[24] R. Waugh, 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/. Last Accessed: 2/22/2019.

[25] B. Haidar, M. Chamoun and F. Yamout, "Cyberbullying Detection: A Survey on Multilingual Techniques," 2016 European Modelling Symposium (EMS), Pisa, 2016, pp. 165-171. doi: 10.1109/EMS.2016.037

[26] B. Haidar, M. Chamoun and A. Serhrouchni, "Multilingual cyberbullying detection system: Detecting cyberbullying in Arabic content," 2017 1st Cyber Security in Networking Conference (CSNet), Rio de Janeiro, 2017, pp. 1-8. doi: 10.1109/CSNET.2017.8242005

[27] I. Ting, W. S. Liou, D. Liberona, S. Wang and G. M. T. Bermudez, "Towards the detection of cyberbullying based on social network mining techniques," 2017 International Conference on Behavioral, Economic, Socio-cultural Computing (BESC), Krakow, 2017, pp. 1-2. doi: 10.1109/BESC.2017.8256403

[28] Noviantho, S.Isa and L. Ashianti, "Cyberbullying classification using text mining," 2017 1st International Conference on Informatics and Computational Sciences (ICICoS), Semarang, 2017, pp. 241-246. doi: 10.1109/ICICOS.2017.8276369

[29] Y. Silva, C. Rich and D. Hall, "BullyBlocker: Towards the identification of cyberbullying in social networking sites," 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), San Francisco, CA, 2016, pp. 1377-1379. doi: 10.1109/ASONAM.2016.7752420

[30] H. Nurrahmi and D. Nurjanah, "Indonesian Twitter Cyberbullying Detection using Text Classification and User Credibility," 2018 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, 2018, pp. 543-548. doi: 10.1109/ICOIACT.2018.8350758

[31] S. Özel, E. Saraç, S. Akdemir and H. Aksu, "Detection of cyberbullying on social media messages in Turkish," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 366-370. doi: 10.1109/UBMK.2017.8093411

[32] A. Mangaonkar, A. Hayrapetian and R. Raje, "Collaborative detection of cyberbullying behavior in Twitter data," 2015 IEEE International Conference on Electro/Information Technology (EIT), Dekalb, IL, 2015, pp. 611-616. doi: 10.1109/EIT.2015.7293405

[33] Y. Chen, Y. Zhou, S.Zhu, and H. Xu (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).

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

[35] R. Pawar, Y. Agrawal, A. Joshi, R. Gorrepati and R. Raje, " Cyberbullying Detection System with Multiple Server Configurations," 2018 IEEE International Conference on Electro/Information Technology (EIT), Oakland, MI, 2018.

[36] Cloud Computing Advantages. Retrieved from: https://www.skyhighnetworks.com/cloud-security-blog/11-advantages-of-cloud-computing-and-how-your-business-can-benefit-from-them/. Last Accessed: 3/4/2019.

[37] Twitter sentiment algorithms benchmarking precision, recall, f-measures. Retrieved from: https://www.linkedin.com/pulse/20141126005504-34768479-twitter-sentiment-algos-benchmarking-precision-recall-f-measures/. Last Accessed: 2/12/2019.

[38] Formspring dataset. Retrieved from: https://www.kaggle.com/swetaagrawal/formspring-data-for-cyberbullying-detection. Last Accessed: 2/22/2019.

[39] Hindi movie reviews dataset. Retrieved from: http://www.cfilt.iitb.ac.in/resources/senti/download.php?get=HPLC.zip. Last Accessed: 2/22/2019.

[40] Hindi tour reviews dataset. Retrieved from: http://www.cfilt.iitb.ac.in/resources/senti/download.php?get=HPLC_tour.zip. Last Accessed: 2/22/2019.

[41] Hindi Newspapers review dataset. Retrieved from: https://navbharattimes.indiatimes.com/. Last Accessed: 2/22/2019.

[42] Marathi tour reviews dataset. Retrieved from: http://www.cfilt.iitb.ac.in/resources/senti/download.php?get=MPLC_tour.zip. Last Accessed: 2/22/2019.

[43] Maharashtra times news review. Retrieved from: https://maharashtratimes.indiatimes.com/. Last Accessed: 2/22/2019.

[44] Twitter API. Retrieved from: https://developer.twitter.com/en/docs.html. Last Accessed: 2/22/2019.

[45] Data imbalance issue and techiques to overcome. Retrieved from: https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/. Last Accessed: 2/22/2019.

[46] English Sentiword net dataset. Retrieved from: http://sentic.net/. Last Accessed: 2/22/2019.

[47] Hindiwordnet dataset. Retrieved from: http://www.cfilt.iitb.ac.in/resources/senti/download.php?get=HSWN.zip. Last Accessed: 2/22/2019.

[48] Google translator in python. Retrieved from: https://pypi.org/project/googletrans/. Last Accessed: 2/22/2019.

[49] Sentiment lexicon for 81 languages. Retrieved from: https://storage.googleapis.com/kaggle-datasets/2473/4130/sentiment-lexicons-for-81-languages.zip. Last Accessed: 2/22/2019.

[50] Marathi abusive words. Retrieved from: http://kaushiklele-learnmarathi.blogspot.com/2013/07/abuse-cursing-swear-words-in-marathi.html. Last Accessed: 2/22/2019.

[51] Scikit-learn. Retrieved from: https://scikit-learn.org/stable/. Last Accessed: 2/22/2019.

[52] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Ijcai, vol. 14, pp. 1137–1145, Montreal, Canada, 1995.

[53] Naïve bayes and Logistic Regression Comparision. Retrieved from: http://dataespresso.com/en/2017/10/24/comparison-between-naive-bayes-and-logistic-regression/. Last Accessed: 2/22/2019.

[54] Why Logistic Regression over Naïve Bayes. Retrieved from: https://medium.com/@sangha_deb/naive-bayes-vs-logistic-regression-a319b07a5d4c/. Last Accessed: 2/22/2019.

[55] Logistic Regression vs Stochastic Gradient Descent. Retrieved from: https://www.quora.com/In-scikit-learn-what-is-the-difference-between-SGDClassifer-with-log-loss-and-logistic-regression/. Last Accessed: 2/22/2019.

[56] T. Christie, “Home - Django REST Framework”, Django-rest-framework.org. [Online]. Available: http://www.django-rest-framework.org/. Last Accessed: 3/10/2019.

[57] M. Robenolt, “Scaling Django to 8 Billion Page Views”, Blog.disqus.com, 2013. [Online]. Available: https://blog.disqus.com/scaling-django-to-8-billion-page-views. Last Accessed: 3/10/2019.

[58] R. Pawar, A. Jangam, V. Janardhana, R. Raje, M. Pradhan, P. Muley and A. Chacko, " Diabetes Readmission Prediction using Distributed and Collaborative Paradigms," First IEEE PuneCon 2018 on International Conference on Data Science and Analytics, Pune, MH, India.

VITA

Rohit Pawar, Computer Science graduate student at Indiana University-Purdue University Indianapolis. I would like to connect with people so, let’s connect on LinkedIn.

PUBLICATIONS

1. R. Pawar, Y. Agrawal, A. Joshi, R. Gorrepati and R. Raje, “Cyberbullying Detection System with Multiple Server Configurations,” 2018 IEEE International Conference on Electro/Information Technology (EIT), Oakland, MI, 2018.
2. R. Pawar, A. Jangam, V. Janardhana, R. Raje, M. Pradhan, P. Muley and A. Chacko, “Diabetes Readmission Prediction using Distributed and Collaborative Paradigms,” First IEEE PuneCon, 2018 on International Conference on Data Science and Analytics, Pune, MH, India.
3. R. Pawar and R. Raje, “Multilingual Cyberbullying Detection System,” 2019 IEEE International Conference on Electro/Information Technology (EIT), Brookings, SD, 2019 (accepted).