CYBERBULLYING DETECTION USING WEAKLY SUPERVISED AND FULLY SUPERVISED LEARNING

ABHINAV ABHISHEK

Bachelor of Electronics & Communication Engineering

Visvesvaraya Technological University

July 2013

Submitted in partial fulfilment of requirements for the degree

MASTER OF COMPUTER SCIENCE

at the

CLEVELAND STATE UNIVERSITY

August 2022

We hereby approve this thesis for

ABHINAV ABHISHEK

Candidate for the Master of Science in Computer Science degree for the

Department of Computer Science

and the CLEVELAND STATE UNIVERSITY'S

College of Graduate Studies by

Thesis Committee Chairperson, Dr. Sathish Kumar, Ph.D.

Department & Date

Thesis Committee Member, Dr. Chansu Yu, Ph.D.

Department & Date

Thesis Committee Member, Hongkai Yu, Ph.D.

Department & Date

Student's Date of Defense: August 2022

DEDICATION

To my father, and my family for their unconditional love and support throughout my journey.

ACKNOWLEDGEMENT

I would like to thank Dr. Sathish Kumar for his continued support. He helped me to learn about different use cases of machine learning and natural language processing. He also guided me to improve my writing abilities. He helped me in every part of my journey through his encouragement and research expertise and great patience.

I would also like to thank my father and family for their sacrifices and support throughout my journey.

CYBERBULLYING DETECTION USING WEAKLY SUPERVISED AND FULLY SUPERVISED LEARNING

ABHINAV ABHISHEK

ABSTRACT

Machine learning is a very useful tool to solve issues in multiple domains such as sentiment analysis, fake news detection, facial recognition, and cyberbullying. In this work, we have leveraged its ability to understand the nuances of natural language to detect cyberbullying. We have further utilized it to detect the subject of cyberbullying such as age, gender, ethnicity, and religion. Further, we have built another layer to detect the cases of misogyny in cyberbullying. In one of our experiments, we created a three-layered architecture to detect cyberbullying , then to detect if it is gender based and finally if it is a case of misogyny or not. In each of our experimentation we trained models with support vector machines, RNNLSTM, BERT and distilBERT, and evaluated it using multiple performance measuring parameters like accuracy, bias, mean square error, recall, precision and F1 score to evaluate each model more efficiently in terms of bias and fairness. In addition to fully supervised learning, we also used weakly supervised learning techniques to detect the cyberbullying and its subject during our experimentations. Finally, we compared the performance of models trained using fully supervised learning and weakly supervised learning algorithms. This comparison further demonstrated that using weak supervision we can develop models to handle complex use cases such as cyberbullying. Finally, the thesis document concludes by describing lessons learned, future work recommendations and the concluding remarks.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABBREVIATIONS

TFIDF – Term Frequency Inverse Document Frequency

SVM – Support Vector Machine

RBF – Radial Basis Function

NLP – Natural Language Processing

RNNLSTM – Recurrent Neural Networks Long Short-Term Memory

BERT – Bidirectional Encoder Representations from Transformers

CNN – Convolutional Neural Networks

CHAPTER I

INTRODUCTION AND BACKGROUND

1.1 Introduction

This thesis work demonstrates methodologies to detect cyberbullying in social media messages using weakly supervised learning and fully supervised learning . Currently mobiles phone and other digital forms of the communication is not only limited to connecting people over voice calls, it is an effective means to share messages and thoughts with either individuals or with entire world. Now we can write our thoughts or opinion in the form of tweets ,or messages and in no time, we can send or share it with everyone. The world seems to be more connected and empowered due to these social media and other messaging services. From deciding on which movies to watch to ordering food based on reviews, these services have made our life simpler and more informative. However, it has also brought many issues with itself such as spread of fake news, hate speech and cyberbullying etc. These issues are making social media an unsafe platform for many people. So, there are research works proposed lately to detect and reduce such messages and news effectively.

The basic idea behind all these works is to remove all these negative impacts of social media, so that people can use it without worrying about these issues. In this work we have taken a very critical issue of cyberbullying detection and performed several experimentations to showcase an effective means to detect the cyberbullying, subject of cyberbullying (gender, religion, ethnicity, and age) and detect if the cyberbullying is a case of misogyny. Our work is intended to detect these messages on different social media platforms so that it can be identified and removed. The first step to handle the cases of cyberbullying is to detect it. This is a complex task as we have millions of messages generated every day and each of these messages have different contextual properties. Therefore, we need a very efficient algorithm to detect such messages correctly. For this we have used the concept of machine learning. Machine learning have played an important role when it comes to extracting useful information such as sentiment, cyberbullying etc. Another important aspect of the machine learning is that it is fully dependent on the data. The data which we use in machine learning consists of two parts: feature set and target label or class. Whenever we have the dataset where both feature set and target labels are present then it becomes easy to sue fully supervised learning and develop a useful algorithm. However, for many domains such as cyberbullying, crime news detection, we have either no or limited amount of data. Therefore, in this work we have used the idea of weak supervision to demonstrate that even with weakly supervised learning we can develop an effective cyberbullying detection.

The rest of the thesis is organized as follows: chapter 2 demonstrate the methodology to train RoBERTa and CNN models using weakly supervised learning to detect cyberbullying. Further it also demonstrates the training of model for multiclass

detection of subject of cyberbullying such as age gender, religion and ethnicity. We evaluated the performance of each model not only in terms of detecting the cyberbullying but also in finding the subject of the cyberbullying and presented a detailed analysis of the results.

Chapter 3 deals with fully supervised learning to detect cyberbullying. In this we have developed three models, the first model is to detect the cyberbullying, second model is to detect if the bullying is based on gender or not and last layer identifies if bullying is a case of misogyny or not. At the end, we have cascaded each model to form a three-layered architecture, which first detects if the text is cyberbullying or not, and if it is bullying then its goes to second layer where it identifies if the bullying is on the basis of gender and if so finally the last layer determines if it is the case of misogyny or not. We have evaluated the performance each model individually and also at the combined three-layered architecture level.

In Chapter 4, we focus on comparing the results obtained using weakly supervised learning and fully supervised learning to demonstrate that even with weak supervision we can build algorithms which can work even better than fully supervised algorithms to detect the cyberbullying.

Finally, in chapter 5 we have presented our lessons learned and future work recommendations along with the concluding remarks.

1.2 Background

Natural language processing (NLP) is an important sub-field of artificial intelligence. We have seen its successful use in sentiment analysis [42]. These uses infers that it has potential to understand the insight of the text deeply as we humans do. Therefore,

it lays a strong ground for its use in other complex domains. Another inspiration for using NLP in more complex domains is the new state-of-the-art algorithms and pretrained model such as RNNLSTM, BERT, distilBERT etc. which has been developed lately. These models have been developed keeping the use case of NLP at the main purpose. Another encouraging fact comes from social media itself as because of the spread of social media there is an availability of huge textual data that can be used for machine learning-based application. So, in this work we have utilized the efficiency of NLP, new state-of-the-art models, and availability of large amount of dataset to detect cyberbullying and contain the cyberbullying messages effectively.

Another issue which we have focused in this work is in terms of collecting large amount of fully labeled data. This is a very common issue of any machine learning application. The labeling of data is a costly and time-consuming work, and so we have used the idea of weak supervision to overcome this limitation. Many related works have been done in this field [9]. But these works either require lot of computational resources or is limited to a particular domain. Therefore, in this work, we have developed a new method to use weak supervision to obtain final labels, which is more informative. Moreover, the method uses simple mathematical equation and requires very less computational resources.

Overall, in this thesis work, we have focused on different aspects of machine learning such as weak supervision, proper model evaluation using different metrics to develop a reliable method to detect cyberbullying using both weakly supervised learning and fully supervised learning techniques.

1.3 Our Contribution

With this work we have made several contributions:

(a) We have built an efficient algorithm using weak supervision and new advanced state-of the-art algorithms such as RoBERTa and CNN, we also developed model to find the subject of the cyberbullying such as age, gender, religion, ethnicity, which provides better details of the bullying.

(b) In the Chapter III we have developed model using fully supervised learning, we have leveraged the efficiency of new advanced models such as RNNLSTM, BERT and distilBERT to build three layers, each layer has different objective, the first layer finds if text is a case of cyberbullying, the second layer detects if cyberbullying is gender based and final layer detects the case of misogyny. So, it does not only detect cyberbullying with this it also finds some more insight about it.

(c) In chapter IV we developed another model to detect cyberbullying using weak supervision and different models such SVM, RNNLSTM, BERT and distilBERT and compared the results of each model to find most efficient model in terms of all measure performance parameters such as accuracy, recall, precision, F1 score, bias and mean square error, to find best model. Also, we have used a different method to calculate final label using "averaged" method. Overall, it provides a simple and efficient method to build model using weak supervision that can utilized for different NLP use cases.

We also compared performance of the models developed using fully supervised learning with those obtained using weak supervision to establish the fact that we can build an efficient algorithm to detect cyberbullying using weak supervision.

Overall, with this work we have presented an efficient algorithm to detect cyberbullying, its subject and the cases of misogyny. The method used in this work can be further utilized in other domains where we have limited amount of fully labeled data.

CHAPTER II

CYBERBULLYING DETECTION USING WEAKLY SUPERVISED MACHINE LEARNING ALGORITHMS

2.1 Introduction

These days social media is not only a way to connect to friends and relatives, but it has grown to an extent that it has become a part of every aspect of our life. Additionally, the Internet has become available to the masses at a very low cost. This development has impacted our society in good and as well as bad ways. As far as benefits are concerned it has empowered people by providing them the ability to raise their voice and share their thoughts in a very convenient way and at the same time it has also opened doors for people to exploit these platforms to harass people. As a result, cyberbullying has now become a very common problem of our modern society.

Although the bullying is ingrained in our life from a long time but the use of electronic means for this had started during 1990's [1], when mobile phones and the Internet were spreading like wildfire. Today it has reached to level that as per study by UNICEF 59% of teens in United States have been bullied or harassed online [2].

Furthermore, as per Google survey, teachers have reported that cyberbullying is on the top in their safety concerns [3]. These recent trends have shown that cyberbullying must be handled as an important domain which is separated from the traditional sentiment analysis domain. Further the need for handling cyberbullying independently also comes from the fact that the texts from the bullying messages can be misjudged as texts with negative sentiments, but we need to also understand that we all have rights to express our dissent about any topic and showing our disagreement must not be considered as bullying. So, cyberbullying must be analyzed separately with highly sophisticated mechanisms.

Besides this rising trend in these cases, the data from the social media platforms has several other challenges. One such issue with the use of unstructured texts is that the social media platforms have limitations of number of words per comment resulting in users trying to express their thoughts in limited number of words, and this further makes it difficult for traditional Machine Learning based approaches like SVM, Decision Tree etc., to correctly identify the intentions behind the messages. Furthermore, there has been an exponential growth in social media users, and it is expected to grow with the same velocity. Hence there is a demand for steady research to enhance the current system to cope with these above issues. The challenges and limitations of the existing system have motivated the research work in cyberbullying detection. Moreover, we have seen many new innovative deep learning mechanisms that have been proposed for Natural Language Processing lately. In this work, we collected data from multiple sources such as IEEE, twitter, YouTube, and Kaggle, collecting data on different aspects based on which the victim was bullied such as age, ethnicity, gender, religion, and other. One major factor in our work was that these datasets were noisy and not annotated properly and only some

percentage of data was cleanly labelled. We leveraged the idea of weak supervision and annotated the data weakly to further process it to build our classifiers [4]. In addition, our objective is to classify data in multiple classes. Therefore, our goal is to design a system which not only classify a data into cyberbullying and non-cyberbullying text but with the second model, we, further able to classify the data to tell if the person was harassed for his age, religion, ethnicity etc. As a result, our objective is to develop a highly effective system which can be further used to find the most common reasons for cyberbullying.

Contributions of this chapter are as follows: (1) We have introduced a way to detect the cyberbullying using weak supervision and new advanced models like Distilroberta and RoBERTa. (2) We have created multiclass classifier for the case where a text is already classified as cyberbullying and can be further subclassified into different classes based on the subject for which the victim was bullied such as age, gender, ethnicity, religion and other (which includes mainly harassment message).

(3) With this work, we have also added main feature selection process that can help to improve the method of applying weak labels. It will also reduce the dependency on domain expertise for applying weak labels and will also give us more liberty in applying machine learning for various other use cases of Natural language processing

(4) Overall, through this research work, we have highlighted the need of considering cyberbullying as a separate critical issue by showing its complexity, importance and the need for new mechanisms to contain it.

Rest of this chapter is organized as follows: Section 2 describes the Background of Cyberbullying, Natural Language Processing and Weak supervision. The related works are explained in section-3 highlighting the details about the methodology and limitations of

the related works. Section-4 presents our contribution details of this paper. In section-5, we describe the main objective of the work and also provides the rationale for this work. Section-6 explains the methodology to apply weak labels using snorkel. Further, this section presents the details about the dataset and about sources from which these data have been collected, it also provides detail about its importance. This section further explains the process of extracting keywords which will be used to generate weak labels. In addition, this section describes the process of snorkel which has been used to apply weak labels to the unlabeled dataset. Section 7 discusses about the experiments and the setup which has been used to perform those tests. In Section 8, we discuss the results for binary classification and multiclass classification experiments. Section 9 discuss future work that should be done to further improve field of weak supervision and cyberbullying detection. Finally, section-10 concludes our overall study.

2.1.1 Background

The rise in electronic means of communications has produced some new means of crime, such as Cyberbullying. It is a form of harassment or bullying using electronic means [19]. In the last decade, it has emerged as a separate entity and is growing rapidly and needs several steps to contain it, Further, it has become easier to harass somebody because today technology has reached everywhere and on top of that all these new social media apps are user friendly and do not require too much knowledge to use it. So, we need to have a series of approaches and our work is a step to help to tackle this critical issue using natural language processing techniques.

Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interaction between computer and human

language [20]. Natural language processing combines the linguistic modeling of human language with statistics, machine learning and we do all these so that computers can process the human text and can interpret its meaning in the same way as human beings understand it [21]. Here our main intention is to train our machine to make them understand the intention and context of the text. Recently, we can see several examples in which machine have been trained to successfully understand and classify the text based on sentiments and context, and in the last few years we have seen models like BERT, RoBERTa etc., which can make machines understand the text more deeply and it can even match the humans understanding. This NLP plays an important part of our work because we need an automated machine, which can scan through millions of texts and then filter out the cyberbullying messages and once, we achieve the required accuracy it can certainly provide us a powerful tool to encounter this issue.

The next important component of our proposed mechanism is the weakly supervised learning., It is a branch of machine learning where noisy, limited sources are used to provide supervision signals for labelling large amounts of data [22]. So, it is an ideal method for us to process data for our work, because social media can generate a lot of data worldwide and has all the necessary variations because it has been collected from people from different regions and thus, they have their own diverse way to express their thoughts. so, now labeling such a large dataset is very critical. Out of many efficient methods which have been proposed lately to resolve the issue of labeling, one very simple and effective method is weak supervision. As a result, use of weakly labelled data has become popular in natural language processing, computer vision, and successful applications in different domains.

Further, when we are working with text, knowing keywords which can help in classifying the data can be advantageous. It results in labeling any amount of data with ease. Along with this, if we use the highly effective snorkel technique which started as a project at Stanford in 2016, further reduces the complexity of labelling large dataset. It may be noted that with weakly supervised learning, the resulting labels are weak and may not be suitable for orthodox machine learning algorithms such as decision tree, support vector machines etc. However, we have seen many new innovations in the field of NLP that gave revolutionary transformer-based Machine Learning techniques like BERT, RoBERTa, DistilBERT. Fusion of new innovation in NLP such as weak supervision and new advanced transformer based models have also shown a commendable result [4].During our study we also found that for sentiment analysis labelling is rather easy as we have plenty of words that can be used to detect negative and positive sentiments, but for a specialized application such as cyberbullying, if we do a little analysis of our data then it makes our weak labels more effective. During our research, we performed some analytics on our cleaned data because we wanted to reach a label where we can even subclassify a text to extract the different reasons for bullying by using weak labels. We created two datasets one for detecting bullying and another to figure out the reasons for it.

2.2 Related Works

Several works have been proposed to detect cyberbullying using machine learning and these approaches include traditional algorithms such as Naïve Bayes, KNN, Decision Tree, Random Forest, and Support Vector Machine algorithms. Weakly supervised learning uses techniques like Embedding, Sentiment, and Lexicon features [5]. This approach is based on orthodox machine learning models, but it has two main limitations

(a) it needs a fully labeled data and to collect such dataset is time consuming and requires domain expertise, (b) although the tradition algorithms like KNN, Support Vector Machine are capable of classifying the text but these algorithms have some limitations when it comes to understanding the context of the text deeply.

Another method was suggested by Islam et al., it uses NLP techniques like TFIDF and machine learning algorithms such as Decision Tree (DT), Random Forest, Support Vector Machine, Naive Bayes to detect the Cyberbullying [6]. This approach has the similar limitations of requiring perfectly labeled data and also it might not be effective in handling complex datasets.

Additionally, another approach was proposed by Muneer et al., which works on predefined keywords and then classifying the tweets in offensive and non-offensive classes [7]. It also has the issue of collecting large amounts of fully labeled data.

A deep learning-based approach proposed by Dadvar utilizes several neural network-based algorithms like CNN, LSTM, Bidirectional LSTM and BLSTM [8] have been tested. The mechanism proposed in this also needs a high amount of fully labeled data and hence has hence doesn't solve the major issue of labeling large dataset.

There is a related work done previously using weak supervision to detect cyberbullying. One such approach was proposed by Raisi et al. which takes the user who sent the message and user who has received it and then it assigns a bully score to the sender and victim score to receiver and then based on the scores it detects bullying [9]. Overall, this approach uses the message structure and the language to find the cyberbullying. One limitation of this work is that it does not show the target of cyberbullying such as age, gender etc.

Overall, the previous works have shown promising results, but the conventional algorithms such as decision tree and Naïve Bayes have their own limitations and needs a proper labelled data to perform well against unseen data. Besides this we see a rapid growth in number of messages being sent by the users and these messages contains lots of variations and it is too noisy to handle with traditional approaches, therefore we need to do further upgrade our current mechanisms.

2.3 Research Objective

The main objective of our work is to provide a very robust architecture which can (a) detect cyberbullying very effectively and (b) then further classify the cyberbullying text into different classes based on the subject, using which the person was targeted. These two tasks were performed using weakly supervised learning to demonstrate the idea that the weak labels might be seen as some random labels criteria, but it has great potential. We also wanted to show that if we do a detailed analysis on the data then we can understand the domains like cyberbullying from data itself and then when we can make use of these learning while labeling the data such that it can produce some great results when we use it with the state-of-the-art algorithms.

Another purpose of work is that we wanted to utilize the latest techniques in the field of Natural Language Processing in different ways to detect cyberbullying. The new transformer-based mechanisms like BERT, RoBerta and DistilBERT are capable of handling any complex task of Natural Language Processing and can understand the context effortlessly, so we wanted to use these capabilities to enhance the ability of our overall system, which is based on noisy labels. Further we tried to make a perfect blend of these

recent innovations with weak labels such that the limitations of weak labels can be handled using the tremendous capabilities of transformer-based models.

Another intention of our work is to use the Snorkeling process more effectively by supporting it with some proper investigation of the cleaned label data, so, instead of using negative sentiment words we created our own list of keywords through count-based analysis and then used these with snorkel to generate weak labels. Therefore, the resulting labels were more informative.

Another objective is to build a multi-class classifier to get some other critical information, which can even divide the text into classes such as age, gender, religion, ethnicity. These classes describe the main subject for reason of cyberbullying. These results can even be used to find statistical data for finding the main targeted subjects.

2.4 Methodology

The proposed mechanism is a blend of several components as shown in Figure 1. It starts with data preprocessing, where data is cleaned to remove unnecessary contents like punctuation, stop words and words having one or two characters only. For weakly supervised learning process, we leveraged the idea proposed by Kai Shu et al. with our own studies and Dataset [4]. One important aspect of the entire system is the selection of algorithms for training we had several options available such as XLNet and BERT-based models like BERT, RoBerta and DistilBERT. We have selected the BERT-based models as these have shown commendable results in other NLP tasks like sentiment analysis, topic modeling and also, we have several implementations of BERT such as RoBERTa. Moreover, it has more improved training methodology and therefore we have selected the RoBERTa pretrained model for our cyberbullying detection approach. Also, we wanted to

verify our approach with completely different architecture. So, we selected CNN model, because it has different approach for classifying the text and on top of that CNN has shown some promising results in the past. The process, dataset, and methods depicted in Figure 1 is further explained in the following subsections.

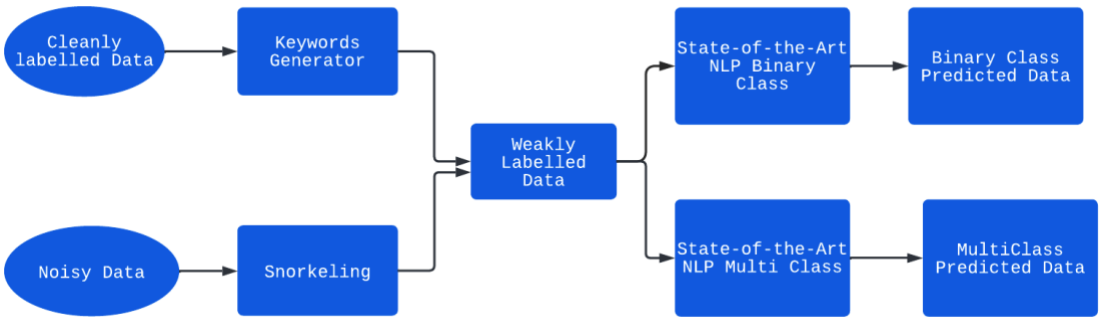


Figure 1: Dataflow Diagram for Overall Process

2.4.1 Dataset

To implement our study, we first collected data from different sources. The data was collected from different sources in such a way that we had a limited number of fully classified data and random data collected from different sources. The fully cleaned labeled data having multiple classes which are age, ethnicity, gender, religion, other and non-cyberbullying dataset was also collected for experimental process. We collected data from different sources such as Kaggle, IEEE. One important aspect of these data is that the cleaned data was completely from different source [10] [43]. These unlabeled data were full of noises and had lots of noise due to platform limitations. Other notable point is that the data from sources such as Twitter has one important defect due to restriction on the number of words people try to express their thoughts or message in limited number of words results into a very unstructured and vague texts. As a result of this, the task of

extracting meaningful information out of these texts becomes tedious, especially when it is going to be processed by machine learning algorithms.

2.4.2 Analyzing data for finding significant words

A major part of our analysis on fully labelled data was to find out the most common words, which were dominant in the text with bullying. The main reason behind this was that cyberbullying is different from the commonly used sentiment analysis, so before using snorkel to weakly label the data, we wanted to make sure that we have a clear understanding of the words and text structure specifically for finding bullying intention. To find such words we took each data of each category: age, ethnicity, gender, religion, other quality in different and then removed all the unnecessary words then we took the counts of each word and plotted them with proper visualization. We found that for each category there were some peculiar words for example for ethnicity we found words such as black or white and other similar words. Using these words, we conducted a detailed study of the patterns of the text from different sources such as Twitter and YouTube. Based on this analysis, we found that we had some vital keywords that helped us to build labelling function in a more precise way. Another achievement of this investigation was that we had a list of conclusive words using which we could generate multiple weak labels for each category. That way, we had sufficient information for multiclass classification.

Further, in our approach we had an important primary task to classify data into cyberbullying and non-cyberbullying classes, so the only thing needed was to combine all the multiple lists into one. With this approach one important gain was that since word lists were generated for every category so after merging these lists into one resulted into a very

robust collection of words to classify the instances into two classes i.e., cyberbullying, and non-cyberbullying.

The next step in our experiment was to generate weak labels for our dataset.

2.4.3 Applying Weak Labels with Snorkel

Once we collected all the key terms the next objective is to effectively generate weak labels for an entire unlabeled dataset for binary as well as multi-class classification processes. To implement the entire process of labeling with ease, we utilized the credible system named snorkel [40]. The labeling process had two major subprocesses, one for binary class and another for multiclass.

2.4.3.1 Binary class

For binary classes, we wrote multiple labeling functions based on different dominant terms. During the process we also tried to find the effect of the number of weak labels on the overall performance of the algorithm. Therefore, for binary classification we created two sets of weakly labeled data. For the first set, only three weak labels were generated and to do so we took all the major words in same list and checked if any words from the list is present in the text. If so, it was labeled as cyberbullying, further two more labels were assigned using regex and one using sentiment polarity of the text. For the second set, 28 labels were created, and each label was created based on one particular word from the lists created earlier.

2.4.3.2 Multiclass

For multiclass classification 6 classes were created with labels from 1 to 5 for age, ethnicity, gender, religion, other and one label for non-cyberbullying. One important point

to note here is that the cleaned dataset was already divided into different categories, therefore we had both clean labeled and weak labeled datasets.

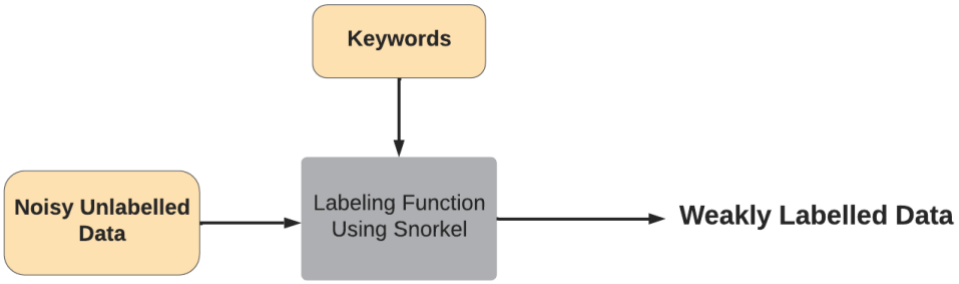
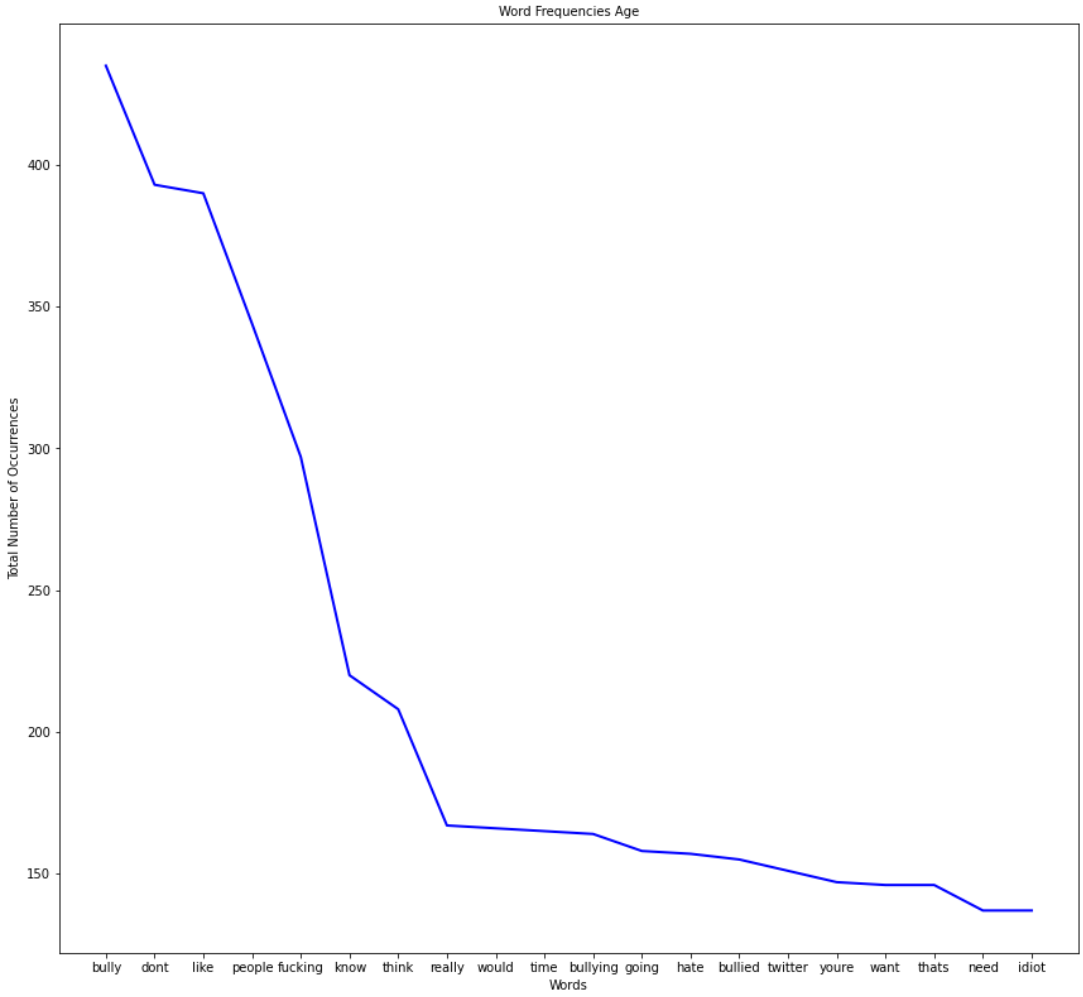


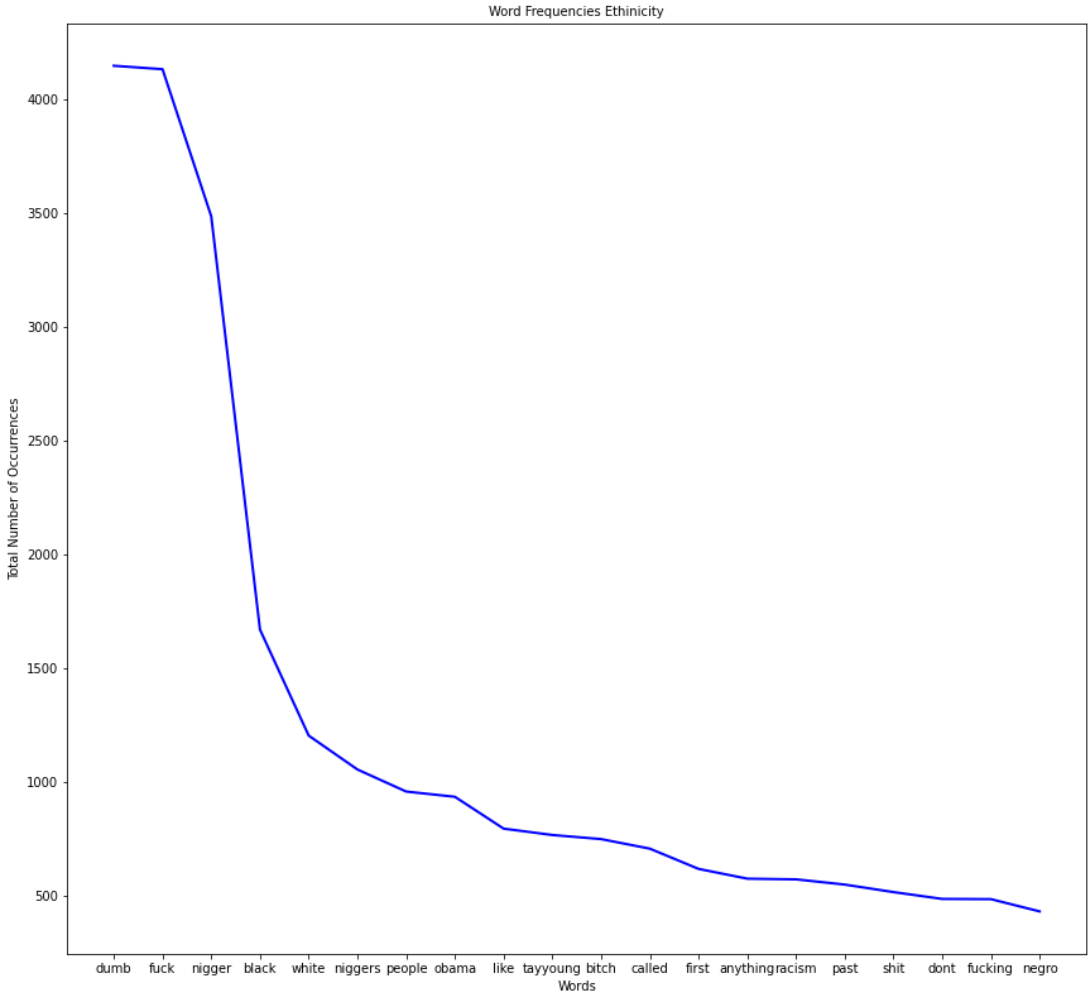
Figure 2: Dataflow Diagram for Weak Label Generation

The Figure 2 shows that how weak labels are applied using keywords generated from cleaned and fully labeled dataset and then using these keywords, we generate labels for noisy unlabeled dataset.

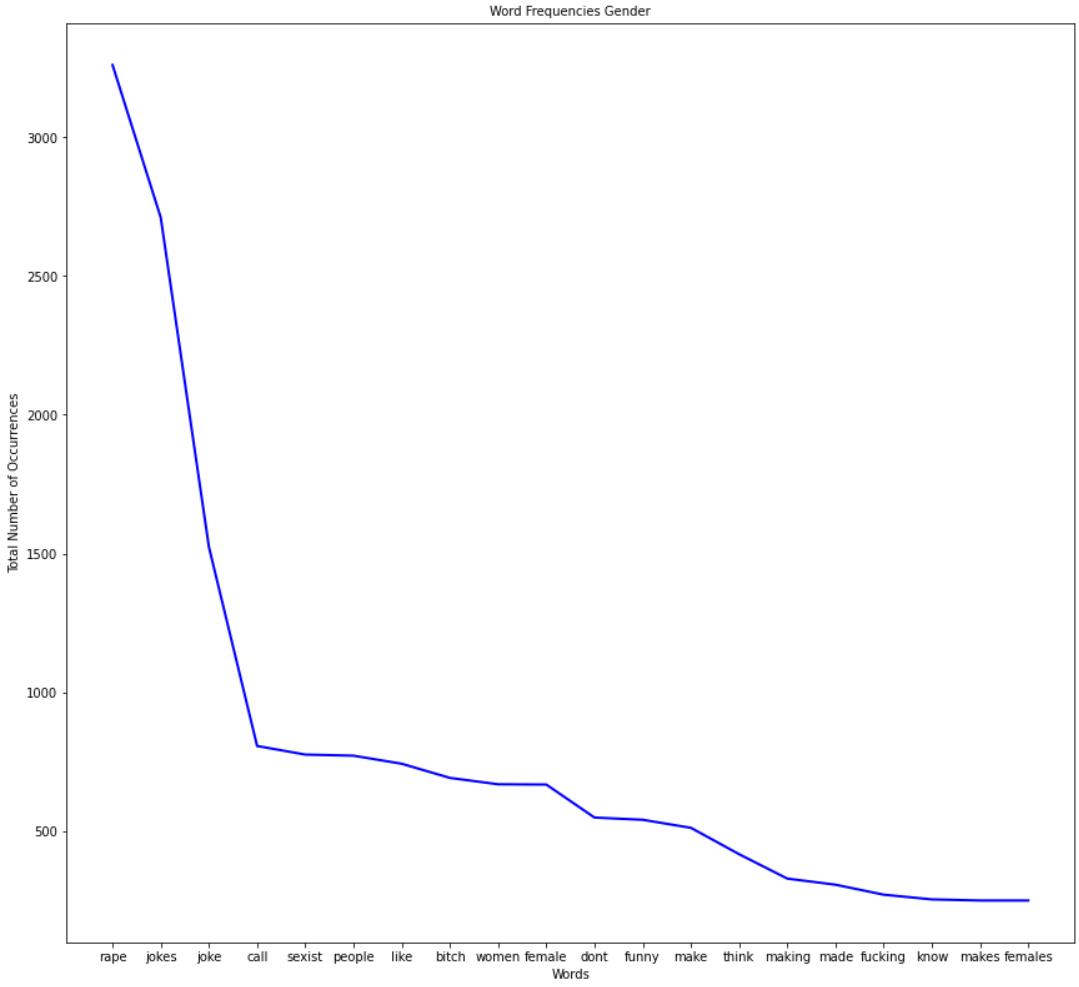
As a first step we need to generate the keywords using the cleaned and fully labelled dataset. To generate the keywords, we have first plotted top 20 words for each classes as shown in Figure 3, such as gender, age ethnicity, religion and then used 5 to 7 words from these top twenty words as keywords to label the noisy dataset.


Word Frequencies Age
Total Number of Occurrences
400
350
300
250
200
150
bully
dont
like
people
fucking
know
think
really
would
time
bullying
going
hate
bullied
twitter
youre
want
thats
need
idiot
Words

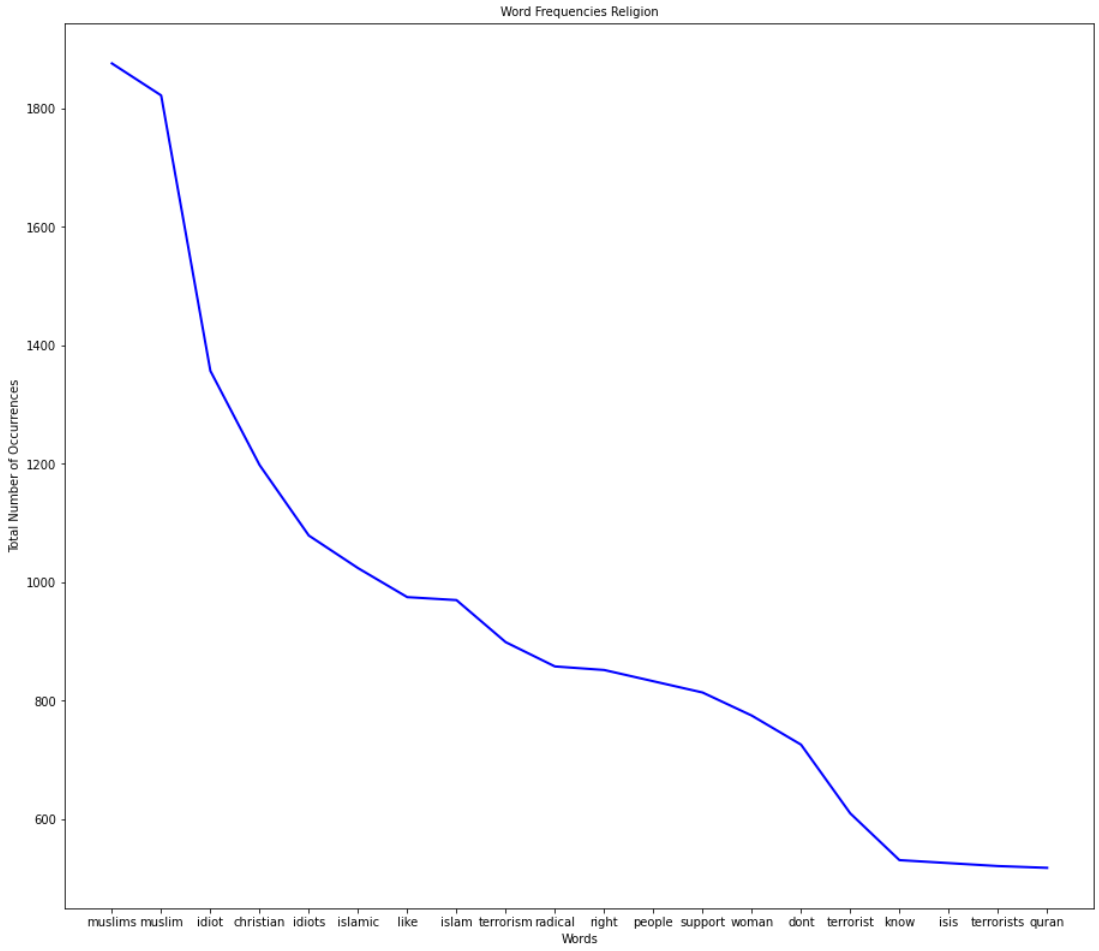
(a) Age


Word Frequencies Ethinicity
Total Number of Occurrences
4000
3500
3000
2500
2000
1500
1000
500
dumb
fuck
nigger
black
white
niggers
people
obama
like
tayyoung
bitch
called
first
anything
racism
past
shit
dont
fucking
negro
Words

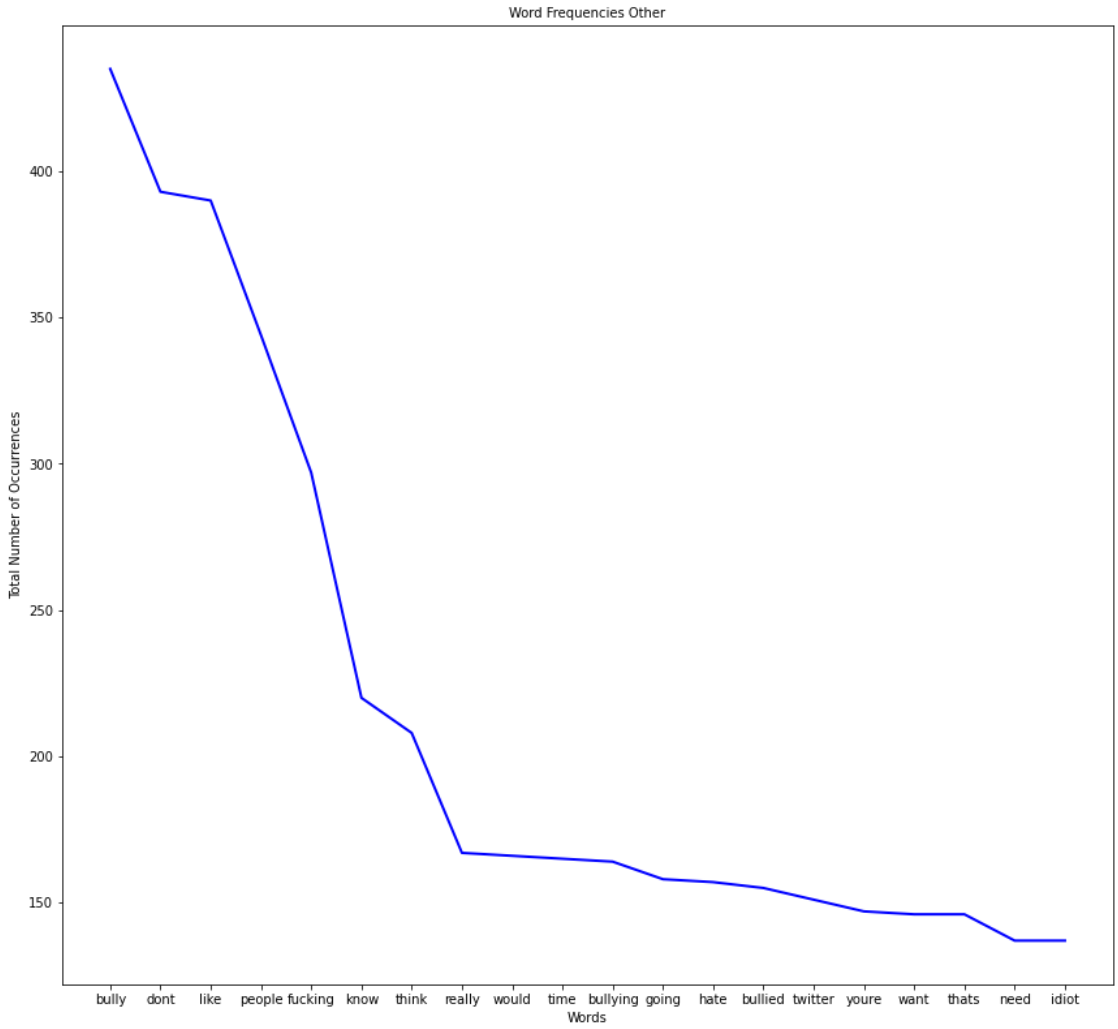
(b) Ethnicity


Word Frequencies Gender
Total Number of Occurrences
3000
2500
2000
1500
1000
500
rape
jokes
joke
call
sexist
people
like
bitch
women
female
dont
funny
make
think
making
made
fucking
know
makes
females
Words

(c) Gender


Word Frequencies Religion
Total Number of Occurrences
1800
1600
1400
1200
1000
800
600
muslims
muslim
idiot
christian
idiots
islamic
like
islam
terrorism
radical
right
people
support
woman
dont
terrorist
know
isis
terrorists
quran
Words

(d) Religion



(e) Other

Figure 3: Frequency Distribution of Words

Figure 3 (a-e) graphs the word count for each class. Figure-3(a) shows the top twenty words in terms of count for the Age class. Figure-3(b) It shows the top twenty words in terms of count for the Ethnicity class. Figure-3(c) shows the top twenty words in terms of count for the Gender class. Figure-3(d) shows the top twenty words in terms of count for the Religion class. Figure-3(e) shows the top twenty words in terms of count for the

other class that does not belong to the rest of four classes (Age, Ethnicity, Gender, Religion).

2.4.4 Model Architecture

Figure-4 illustrate the optimum machine learning architecture adapted from related work on using weak label for fake news detection [4]. The main rationale to adopt the particular architecture is that it has presented a highly generalized mechanism, which tries to fully utilize the information provided in the data from different sources even though it is full of noisy signals by using a limited number of cleaned labels.

Following is the mechanism we have used in our work.

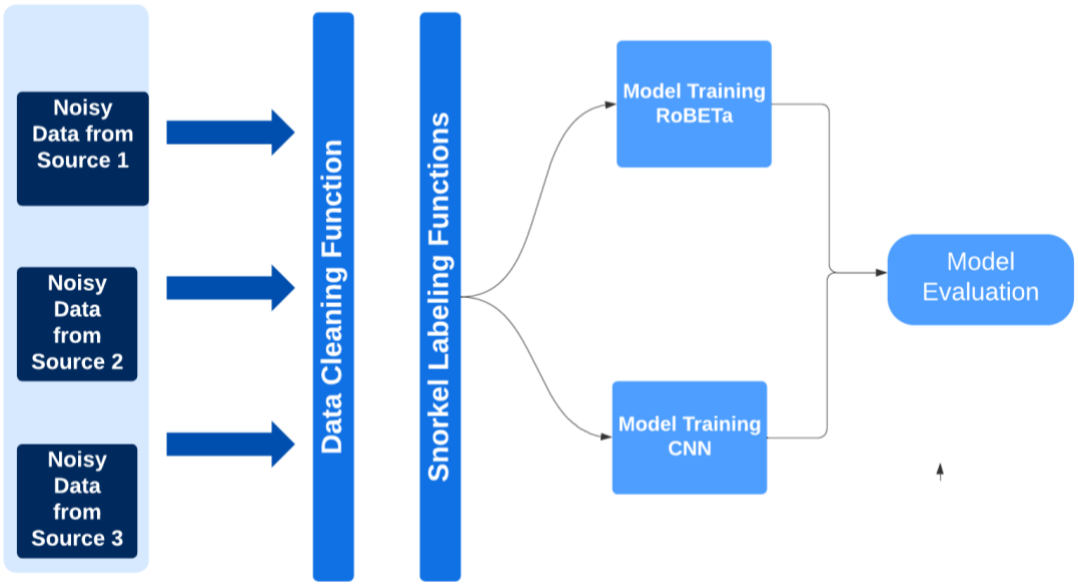


Figure 4: System Architecture

Figure-4 shows the overall structure of our methodology, the top layer is the data collection and preprocessing step. In this layer we also performed analytics to generate

keywords. The next layer shows the process of snorkel. In the final layer, the model is trained with all this data.

2.5 Experiments

In this section, we explain our binary classification and multi-class classification experiments. In binary classification experiments, the model is trained on weak labels to predict if a text belongs to cyberbullying or not. The second experiment is to build a model that can take a text which is already classified as cyberbullying text and predict the subject of cyberbullying such as age, ethnicity, gender, religion, and other.

In this section, we explain about our experimentation setup for binary and multiclass classification and the metrics to evaluate the success of our experiments.

2.5.1 Binary Classification

We performed model training for binary classification (cyberbullying and non-cyberbullying). We combined the clean and weak labeled data for our experiments. We took the first set with three weak labels for the initial phase. For getting embedded data for training, we used 256 tokens. The very first experiment was done with a highly robust **distilroberta** model with a learning rate of 1e-4 and batch size of 16. One important part of this experiment is **distilroberta** model**. Distilroberta** model is a distilled version of the RoBERTa-base model, which has 6 layers, 768 dimensions, and 12 heads, totalizing 82M parameters. This is a very fast and highly effective pretrained model.

Further, we have collected cleanly labeled data and mixed it with weakly labeled data. Out of around 90,000 points data, around 20% were cleanly labeled data and the rest were weakly labeled data. The training worked perfectly and gave some excellent results as shown in Table 1.

For the second experiment, we took RoBERTa and RoBERTa and we tested that with only three weak labels this is an important experiment because during training we found that for the dataset having only three weak labels the algorithm did not perform well and gave only 45% accuracy. But when we increased the number of weak labels the accuracy increased, and we got very promising results and further for the dataset having multiple (28) weak labels it performed exceptionally well. As generated weak labels are increased, the accuracy for the classification unit improves, these results demonstrate our thesis that if we perform data analysis and then generate weak labels then as we increase the number of weak labels the accuracy will improve, because each label will add some information to the learning of the machines.

For binary classification, we performed 4 experiments. Out of 4, three experiments were done by taking both cleanly labeled and weakly labeled data. Here the majority of dataset consists of weakly labeled data and these datasets were fully unstructured, but during the training process we found that our research done for the domain together with our method to generate weak labels really helped us to achieve excellent results.

Since we got outstanding results, we performed another experiment with only weak labels, the main reason behind the last experiment with only weak labels is to test the two main concepts : first, we wanted to find is that the model architecture is capable enough to handle the training with only weak labels and second we wanted to find that until which extent the preprocess layer we created before applying snorkel has helped the machine to learn the text more accurately. After training, we found that at first the model architecture is very efficient and use of pretrained models such as Distilroberta really helped us to

achieve our goals. On the other hand, our preprocessing layer too helped us to improve the overall learning.

Therefore, overall experimental analysis for binary classification was very much fruitful and gave us incredible results. And these results gave strength to our overall study. Overall results for the binary classification results are summarized in Table 1.

2.5.2 Multiclass Experimentation

The performance achieved in the binary classification encouraged us to go one step further and experiment the multiclass classification learnings. The main setup for experiment was to arrange the dataset in the proper way, so that keyword for weak labels is created using clean data and then applied it to noisy dataset collected from different sources. Then we trained two models one with RoBERTa and another with CNN and these two experiments were done with both weakly labeled and cleanly labeled data. We took a Convolutional Neural Network and distilbert pretrained model and results of these experiments are promising.

2.6 Discussion of Results

The experiment results are encouraging and are discussed as below.

In this section, we discuss the results obtained for binary classification and multi-class classification

2.6.1 Binary Classification

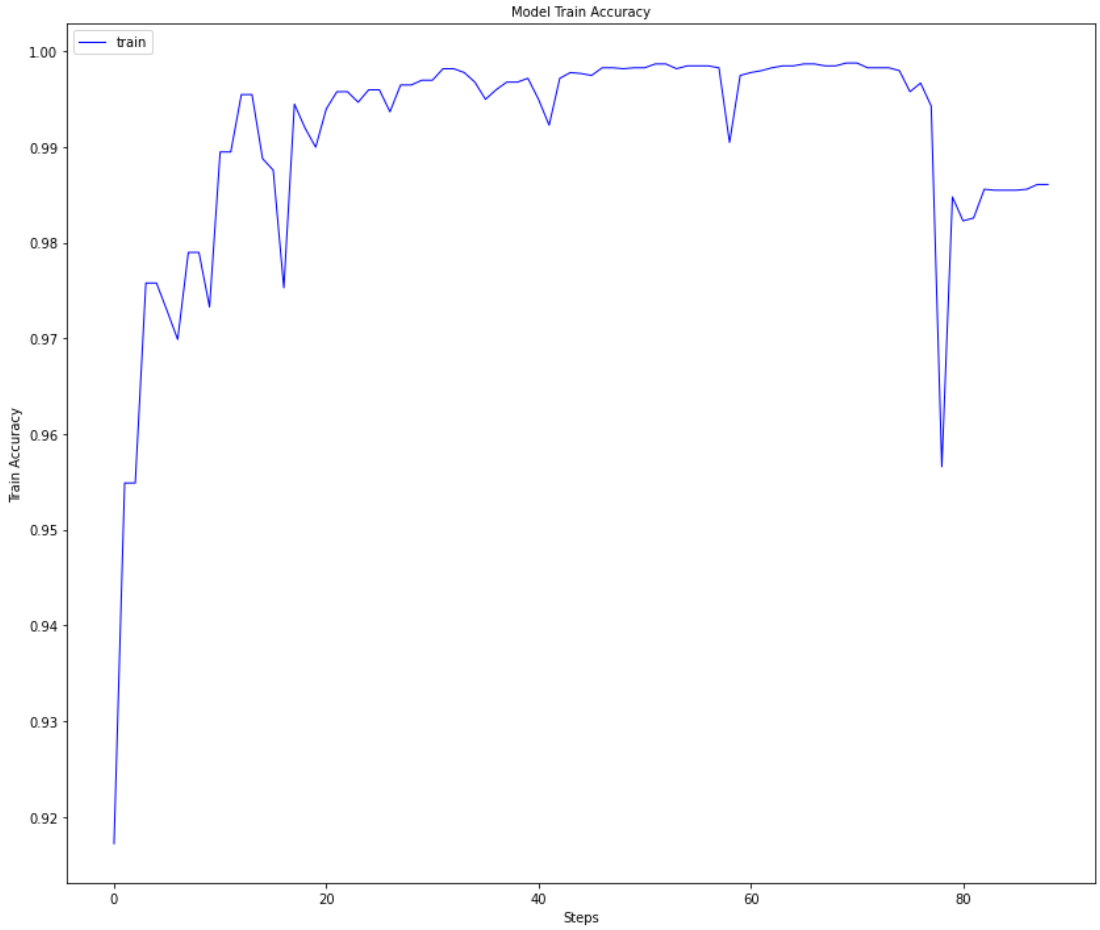
One of the metrics, we used to measure the performance of our approach is F1 score. The F1 score combines the recall and precision by taking the harmonic mean of precision and recall. F1 score is very useful in case of imbalance data and in our dataset, we found some imbalance. Therefore, we decided to use this metric. Moreover, for our

case we are concerned for false Negative and false Positive cases, because we did not want to classify something as cyberbullying as non-cyberbullying and vice versa, and F1 score is best for our purpose. F1 score is calculated as follows:

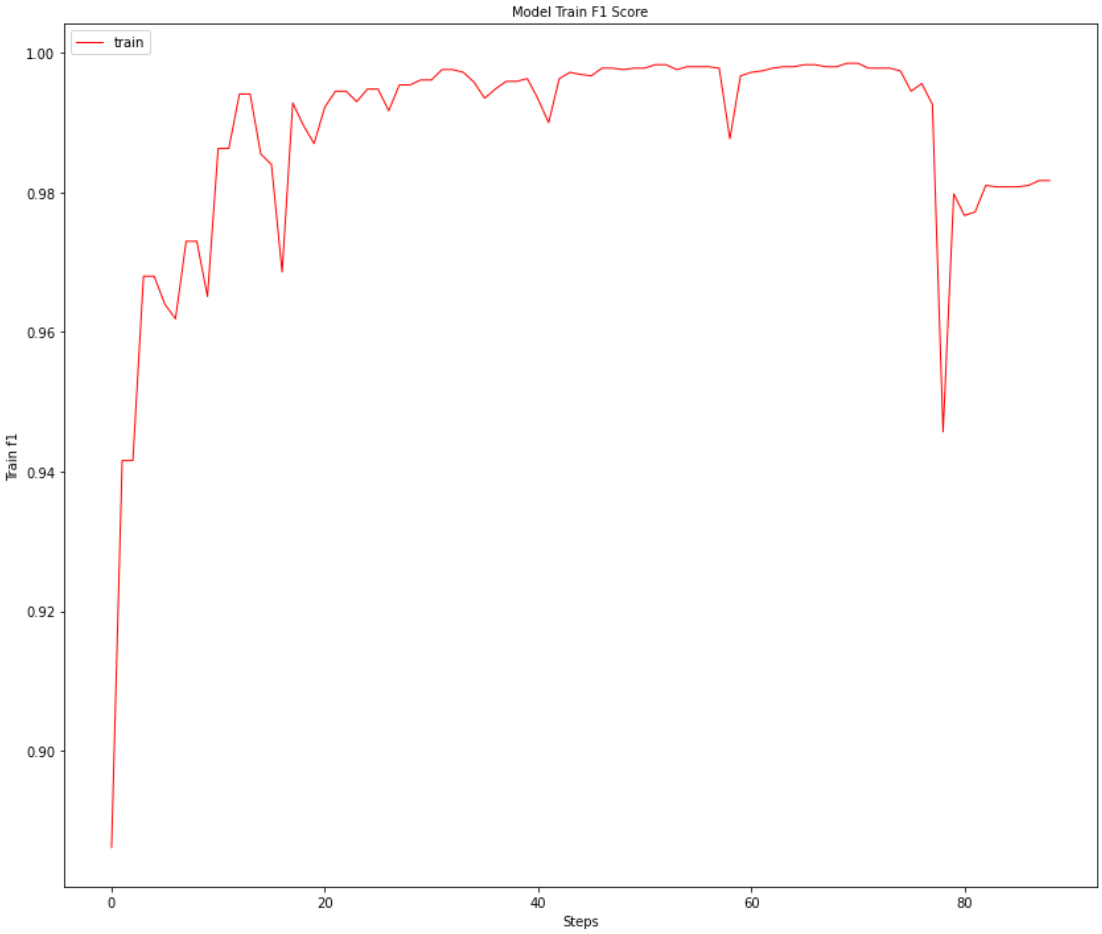
F1 score = 2*(True Positive)/ (True Positive + False Negative) …… (1)

Another metric we used for evaluating the model performance is accuracy . We also verified the model after increasing the amount of weakly labeled data and the noise data and found results are shown in Table 1.

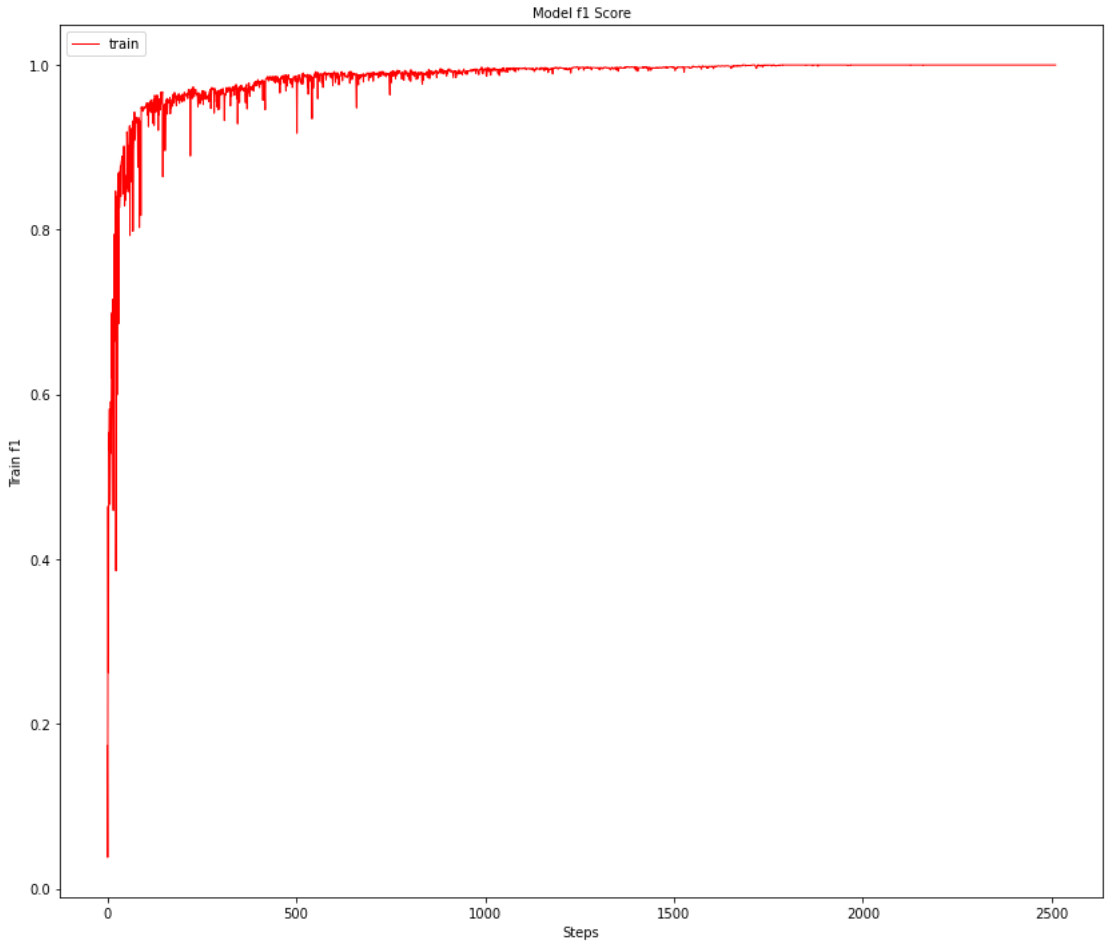
To evaluate the performance of the approach, we collected the F1 score, loss and accuracy at each step and plotted it for better understanding of the training process and carefully observed the performance for our binary classification and multiclass classification.



(a) Results of Binary Classification Experiment1- Accuracy Score for RoBERTa



(b) Results of Binary Classification Experiment1- F1 Score for RoBERTa



(c) Results of Binary Classification Experiment 1-F1 score for CNN

Figure 5: Binary Class Experiments Results

In Figure 5, the Y-axis shows the value of accuracy and F1 score and X-axis shows the steps count.

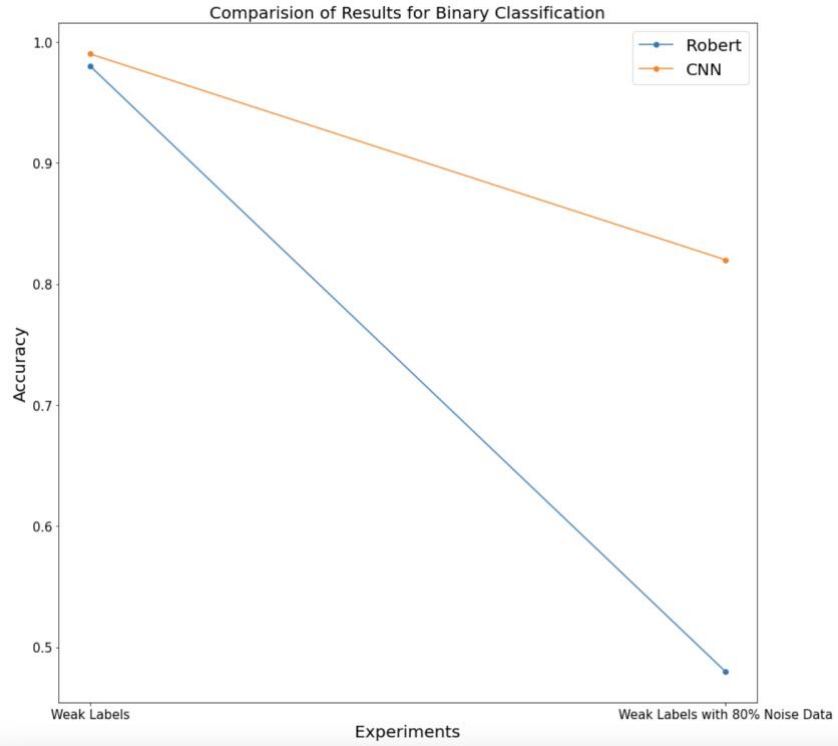
The Figure-5(a-c) shows the experimental results for binary classification experiments, the Figure-5(a) demonstrate the accuracy for RoBERTa and Figure 5(b) shows the F1 Score for the RoBERTa model where we have both cleaned and weak labels. The Figure-5(c) shows the F1score for CNN model, with both cleaned and weak labelled dataset.

The results for binary classification experiments are shown in Table 1.

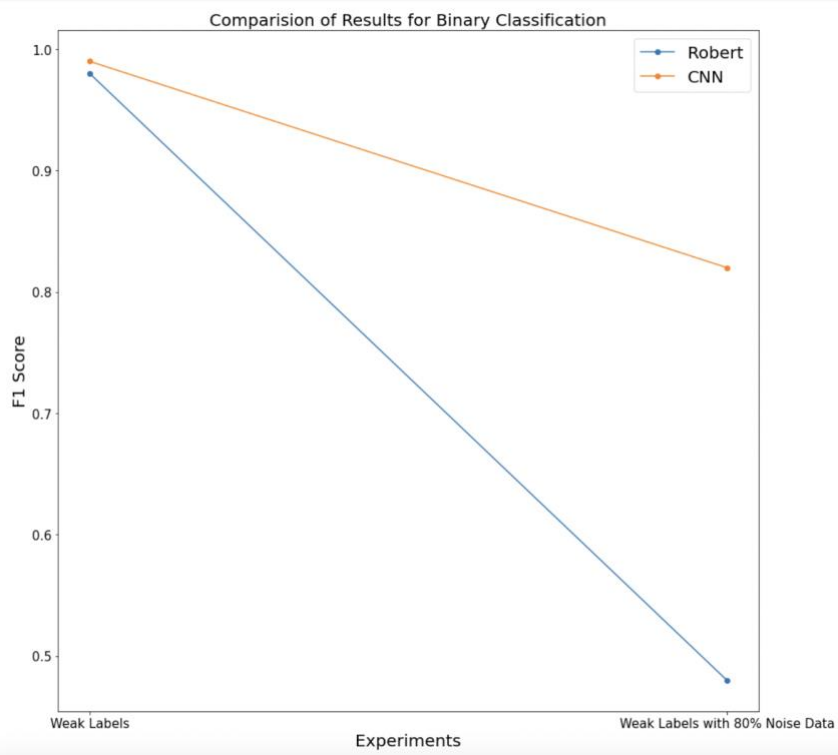
Experiment	Accuracy	F1 Score	Loss	Model
Weak+clean Labels	98%	98%	7%	RoBERTa
Weak+clean Labels(Increased Noise Data)	48%	48%	80%	RoBERTa
Weak+clean Labels	99%	99%	3.7%	CNN
Weak+clean Labels(Increased Noise Data)	82%	82%	6.4%	CNN

Table 1: Binary Class Experiments Results

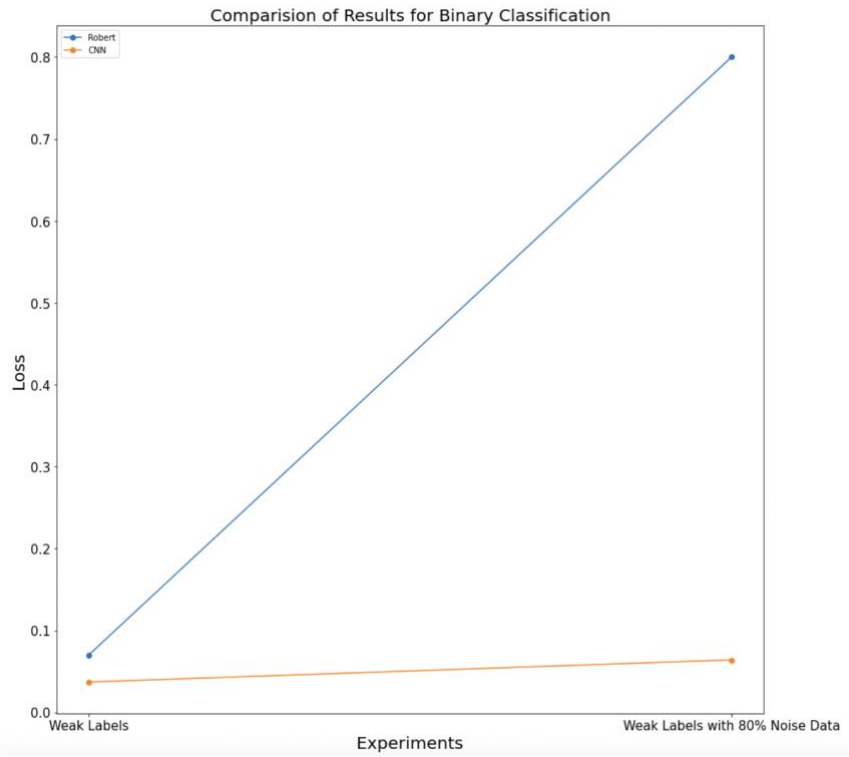
The first row shows the result of the training process for classifying data as cyberbullying and non-cyberbullying taking both weakly labeled and cleanly labeled data for RoBERTa model. The second row shows the training results for RoBERTa model considering the noisy data. The third row shows the training results with both cleanly labeled and weakly labeled data using CNN model. The last row shows the results for CNN model with higher number of nosy data.



(a) Binary classification Accuracy Comparison for RoBERTa and CNN



(b) Binary classification F1 Score Comparison for RoBERTa and CNN



(c) Binary classification Loss Comparison for RoBERTa and CNN

Figure 6: Evaluation of Binary Classification using Accuracy, F1, and Loss Comparison for Roberta and CNN

The Figure-6(a) shows the performance evaluation using Accuracy, F1 score and loss for RoBERTa and CNN models for two cases: (i) for the clean and weakly labelled data with hardly 20% noise data and (ii) weakly labelled data with almost 80% noisy data. As shown in Figure 7, RoBERTa did not perform well compared to CNN model.

For the binary classification, with the RoBERTa we have got 98% accuracy and F1 score of 98%. However, when we increased the percentage of noisy data the result were not satisfactory for RoBERTa we have got 98% accuracy and F1 score of 98%. However, when we increased the percentage of noisy data the result were not satisfactory for RoBERTa we have got 98% accuracy and F1 score of 98%. However, when we increased

the percentage of noisy data the result were not satisfactory for , and accuracy did not improve after 48%. On the other hand, the CNN model worked really well in both cases, with 99% accuracy and 99% F1 score for both weak and clean label , and accuracy did not improve after 48%. On the other hand, the CNN model worked really well in both cases, with 99% accuracy and F1 score for both weak and clean label , and accuracy did not improve after 48%. On the other hand, the CNN model worked really well in both cases, with 99% accuracy and F1 score for both weak and clean label. For increased noisy data, accuracy and F1 score fell slightly to 82%.

One important rationale for these improved results in spite of weakly labeled approaches are as follows:

(a) we have selected the keywords after some proper analysis, therefore, although the labels are weak and random, it has relevant information contained in it and on top of it there are multiple labels and each label is a result of our analysis, and as a result, the accuracy increased due to combination of all these factors.

(b) The second reason for better performance is that we have used highly robust pretrained models which further enhanced the efficiency of the mechanism.

2.6.2 Multiclass Classification

The other main component of our experiment is the multiclass classification. Here, we tested the same architecture with multiclass labels. These labels were also weak and from the keywords used in first experiment. The results in terms of accuracy and F1 scores were encouraging. For the Roberta pretrained model, we got an accuracy of 86% and F1 score of 86% considering weakly labeled and clean labeled data. Also, for CNN the

accuracy and F1 score slipped to 82%. Table 2 summarizes the results for multiclass classification, with Roberta and CNN considering both clean and weakly labelled data.

Experiment	Accuracy	F1 Score	Loss	Model
Multi Class clean + weak	86%	86%	42%	RoBERTa
Multi Class clean + weak	82%	82%	70%	CNN

Table 2: Multiclass Experimentation Performance Metrics

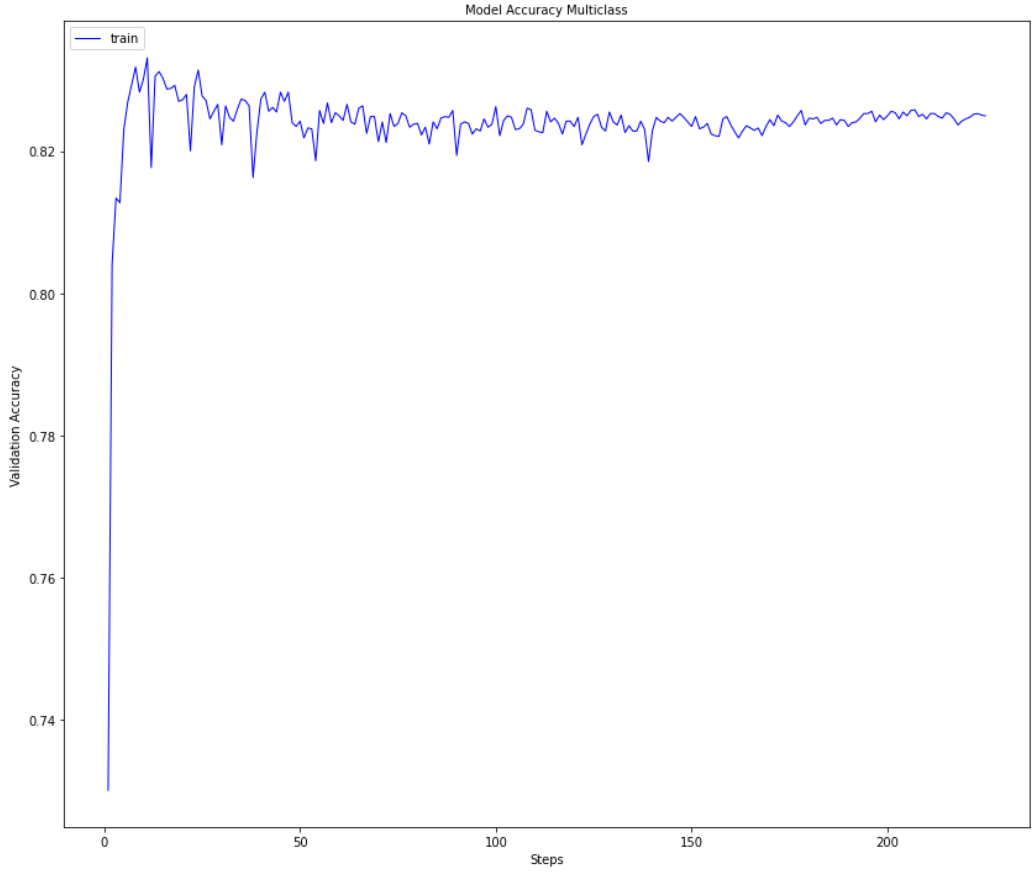


Figure 7: Multiclass Classification With CNN- Accuracy

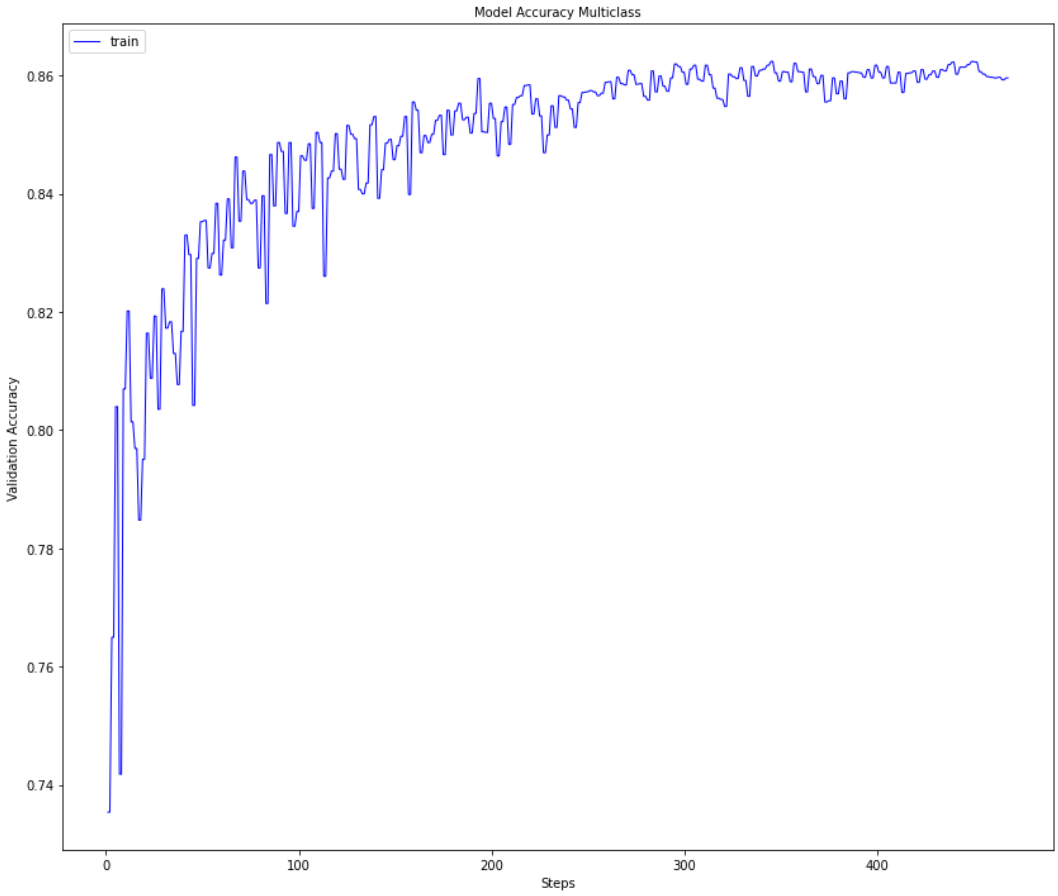
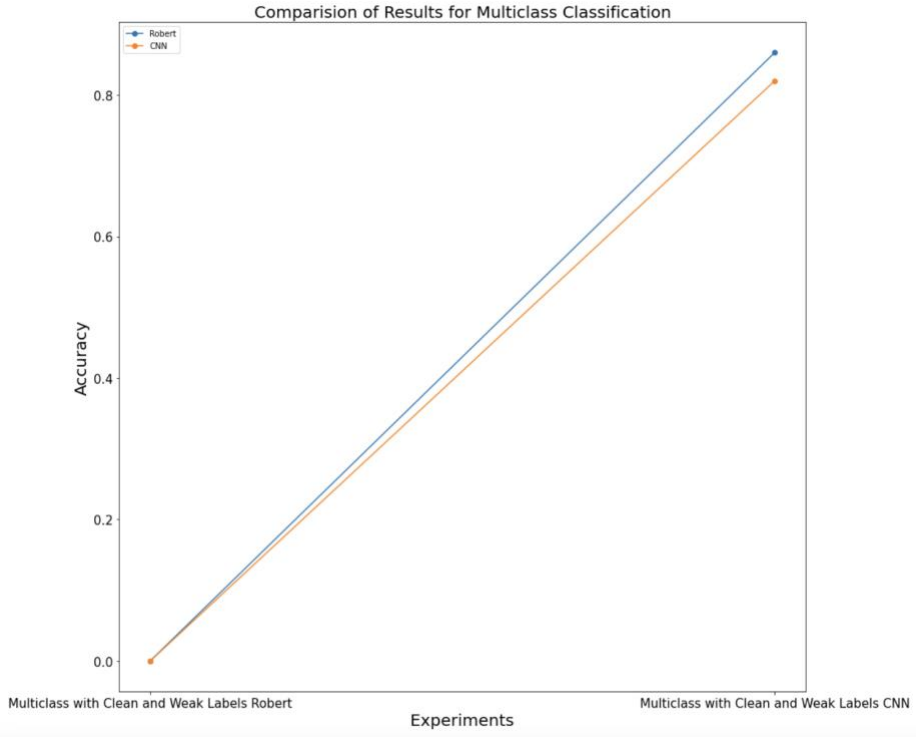
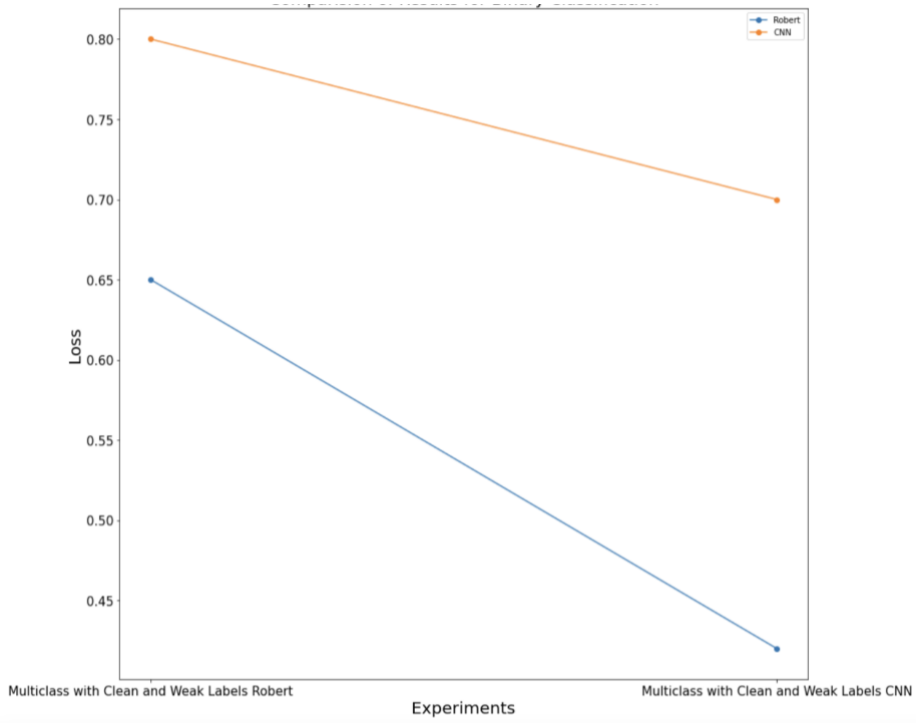


Figure 8: Multiclass Classification With RoBERTa- Accuracy



(a) Multiclass classification Results -Accuracy



(b) Multiclass classification Results - Loss

Figure 9: Experimental Results for Multiclass Classification.

The Figure 9 shows the comparisons of results for RoBERTa and CNN model with 20% noisy data. The Figure 9(a) shows the Accuracy and F1 score comparison and 9(b) shows the comparison of loss obtained using RoBERTa and CNN models. Both the models have similar results accuracy and F1 score, but for loss the CNN has much lower loss than RoBERTa. So, over all CNN worked better than RoBERTa.

Therefore, due to these encouraging results, we were able to build two classifiers with the same dataset with our proper analysis and utilizing the highly generalized model architecture.

2.7 Research Contributions

Following are the detail research contributions of our work:

(a) In this work, we have addressed the issue of unavailability of large amounts of fully labeled data. This have been a fundamental issue in machine learning, especially when we need to apply ML for some specialized cases such as cyberbullying. As a result, we certainly need to have some highly sophisticated tools to encounter this issue, since it becomes very difficult to find huge dataset, which is fully labeled data and it will need a lot of hard work and time just to manually label data. So, we implemented the idea of weak supervision in our experiment. Also, in order to use weak supervision for our work, we needed to take a different approach. This is because the functionality of weak supervision depends on applying the weak labels to the data, and to generate labels we needed some criteria and one such criteria is to check polarity of the text. However, this approach has two major issues (i) it would have just provided us with one weak label and during our work itself we found that even if we used only three labels the accuracy is very low, so we needed some more labels, and (ii) If we only consider only the polarity, we actually depend

on the sentiments of the text and for our case this was not enough because we wanted build an algorithm which should not be dependent on the sentiment and have a proper learning of the domain to detect the cyberbullying. So, to handle this issue we needed a novel approach described as follows:

We added a layer of text analysis before applying the weak labels to the dataset and this step has given us a fair number of keywords and these keywords are very critical because it gave us a good understanding of the texts that belong to cyberbullying class. This approach has another aspect, i.e., when we train a machine then we make them learn to understand the overall text, so we need to have some words which should have some ability to throw some light on the texts. So, here we applied two important methods, (i) understanding of the domain and (ii) we took information from our data analysis step to generate a series of labels.

(b) Using this approach, we made another contribution of building multi-class classifier to understand reason for cyberbullying such as age, gender, ethnicity, religion and others. There is one class named as “other” which contains mainly threat and harassment messages. To do so we utilized our detailed study of data to extract important information and then used this learning to effectively classify a large set of noisy data which were not even prepared to be used for multi class classification.

(c) Through our experiments we also demonstrated that if we do a little investigation on data before applying weak labels then we can use the weak supervision to many other domains effectively. After verifying our learning using extensive experimentation, we got encouraging results to upheld the fact that with weak supervision we can solve one of the most critical problem of Machine Learning which is data

annotations for different domains. It not only reduces the efforts needed to label data but during our experiments we found that if we do a little examination of data, we can easily find the crucial words or information using which we can even go further and build different classifiers that can help us to get minute details even with a limited domain expertise.

2.8 Future Work

Our system is very effective in handling the issues of unlabeled data. As a future work, the proposed architecture should be tested with other NLP use cases to demonstrate its robustness. Further, experiments can be done to perform initial data analysis before applying weak labels to effectively select proper labelling functions. Also, we found that although snorkel is a very effective tool to ease labeling procedure, we certainly need some analysis of how we can select criteria to decide weak labels. This analysis is very important because if we want to use the concepts of weak supervision in other domains, where we may have a very limited amount of data. Therefore, if we can improve the process of labeling , we will be able to open up a whole new scope for weakly supervised learning.

2.9 Conclusion

Social Media has provided us with the ability to express our thoughts with the entire world but at the same time we also need to have an advanced system to recognize and stop harassments in this platform. Our method provides one effective way to detect cyberbullying from a wide range of texts with ease. The proposed technique also allows us to encounter issues of noise and variations, which is very common issue in these types of unstructured casual texts.

During our experimentation with binary classification to determine cyberbullying and non-cyberbullying classes, the CNN worked better than RoBERTa. On the other hand for multiclass classification i.e., for determining age, gender, religion, ethnicity classes the RoBERTa worked better than CNN. Further, we observed that with both the models we got promising results, which further solidify our concept that with the combination of weak supervision and applying new advanced algorithms, we can develop reliable methods for solving many NLP use cases.

Based on our experimental results, we also conclude that weak supervision has a lot of scope and has the ability to be applied in many different domains. Further, the new state-of-the-art machine learning models make the weakly supervised machine learning more efficient and thus the improvements in weakly supervised learning-based model improvements has the potential to advance the field of machine learning.

CHAPTER III

CYBERBULLYING DETECTION USING FULLY SUPERVISED MACHINE LEARNING ALGORITHMS

3.1 Introduction

The rapid increase in the implementation of machine learning algorithms has created a necessity to further analyze the reliability of the algorithms and then develop different methods to improve it. One such aspect of the machine learning algorithm, which must be evaluated before applying it to a certain domain is fairness of the machine learning algorithm. The term fairness of machine learning algorithm basically refers to various methods to improve the model's decision-making capabilities in terms of bias. In technical terms suppose we have a model A, which generates predictions R, then A must be statistically independent of R. In other words, for our use case of cyberbullying detection we have 2 classes cyberbullying and non-cyberbullying so that any text X will have equal probability of being classified by the classifier in each of the classes i.e. cyberbullying and non-cyberbullying. Further to understand it in simple words, when a trained model is being used to predict values and if the predictions are systematically unfair to a particular group, then we say that model is biased [26].

The importance of measuring the reliability of the model becomes more critical when we want to utilize the models for crucial decision-making process, because reliability is one of the most important factors which decides if the developed model can be used for some critical application or not. Due to its importance, there have been several works done in past to first properly investigate the important factors that affect the model's fairness the most and then we have seen several works that have been done to resolve those factors. Out of many factors, two important decisive factors are data and algorithm selected for training, because these are two major sources from where the models get bias, and it also affects the overall performance of the model [29]. Any major imbalance in the dataset, like for binary classification one class has 10000 records and other has only 2000, will mostly make the trained model biased towards one class, such that if we have not performed data preprocessing properly, then we lose certain pattern in features mapping and thus it effects the outcome of the model.

Another source of bias is selection of proper algorithm for the required task. For instance, taking NLP as an example, we can say that the text contains lots of information which are important in getting sentiments, context, threat intentions, harassment intentions. But at the same time, it is also required that our model must be capable of extracting those useful contents from those texts, and if model architecture is too simple, it may fail to understand those features that can introduce a lot of bias in the model.

Therefore, in this work our main focus is on model bias and its efficiency from a supervised leaning context. We have demonstrated that how by selecting more advanced model architecture we can improve the fairness of the model. To demonstrate our approach, we have selected the key issue of cyberbullying detection, and to evaluate our approach,

we have experimented with efficient models such as SVM (with different kernel), RNNLSTM, BERT and DistilBERT. In our process architecture, we have created three different layered approach (i) the first layer detects that if a given text is a case of cyberbullying or not, (ii) the second layer detects if the bullying was done based on gender or not and (iii) finally the last layer predicts if the target or victim of cyberbullying was a woman or not. We have trained several models and then compared the accuracy, recall, precision, F1 score, bias and mean square error to establish the learning of our work.

Rest of the chapter is organized as follows: Section 2 describes the related works in the field of cyberbully detection, and fairness of the algorithms in Natural Language Processing. Section 3 explains the methodology including data collection, data cleaning, data preprocessing, along with details of all the algorithms and model training process for each of the layers. In section 4, we have presented a detailed analysis of experimental results in terms of performance and fairness of models for each of the layers. Section 5 discusses about future work recommendations. Finally, section 7 concludes our overall study.

3.2 Related Works

Several works have been done in the past to examine the fairness of the different algorithms; one such prior work have been done using Causal Bayesian Networks (CBNs) [44]. This is a simple and useful tool that can be used to find different possible data-unfairness scenarios. CBNs can also be used as a powerful quantitative tool to measure unfairness in a dataset and to help researchers develop techniques for addressing it. The tool is very effective when we consider the unfairness coming from the data. However, one

limitation of this work is that it dealt with correcting fairness from data, but it does not include how fairness gets affected from the model architecture related issues.

Another work that has been done in NLP in medical domain is based on medical datasets which are structured and unstructured. The study examines True Positive and True Negative rates on clinical prediction task between different protected groups. It also uses multi-model architecture for different data variants i.e., structured and unstructured (CNN, bi-LSTM) [28].

One survey work that has been proposed by Mehrabi et al. it finds two sources of the bias, first due to data and second due to algorithms. The work also considers the several cases from real world where unfair machine learning algorithms have led to suboptimal and discriminatory outcomes [29]. Also, as per our knowledge there is no work proposed with combination of English and Spanish for cyberbullying detection.

Overall although some work has been done to find fairness from data, there are plenty of scope of new studies to further evolve the techniques of finding sources of unfairness from data as well as from model architecture, while the existing works are concentrated around different domains of NLP such as crime news analysis, fake news detection, cyberbully detection can be explored further.

Author	Descriptions	Dataset	Limitations of the solutions	Any other relevant information
Luca Oneto et al.	Authors have presented an approach to learn fair representations that can generalize to unseen tasks.	Wine quality	It is mainly concentrated for bias coming from data and does not include the algorithmic aspect.	This work also demonstrate technique that can be used for legal restrictions for the use of sensitive attributes.
John Chen Et al.	It focuses on the fairness of a multi- model approach on medical dataset. One model is based on structured data, and another is on unstructured data.	MIMIC-III that contains data associated with 60,000 intensive care unit (ICU) admissions	It is based on the medical domain and needs to be implemented in different domains.	It also includes logistic regression application on the output binary classification probabilities from the previous models.
Mehrabi et al.	This is a survey report that has investigated many real-world applications that have manifested biases in different ways.	It is survey paper with multiple methodology with no dataset.	It is survey paper and thus it needs to be analyzed and ideas in this can be tested with different domains.	This work has also provided a summary of the various biases' sources that could influence AI applications.

Table 3: Description of Related Works and Their Limitations.

As shown in Table 3, the prior works are mainly on mostly on the bias coming from the dataset. As per our knowledge, there are not many related works that evaluate the bias from algorithm perspective. So, in our work we have used the state-of-the-art algorithms to build a fair model, which can handle bias in the data. Further, existing work has been

done with orthodox model algorithms. However, in our work we have used latest highly advanced and pretrained model to build a fair model with even noisy data as an input.

3.3 Methodology

Scope of our system includes three layers: layer-1, which detects if the text is cyberbullying or not. Layer-2 which detects that cyberbullying subject is “Gender” based or not, and layer-3 which detects if it is a case of misogyny. Considering the fact that each layer has their own functionality, we have carefully selected data for each layer.

Further the system is designed in such a way that we could be able to verify our hypothesis that using the more complex and latest learning architecture, we can reduce the bias and increase fairness of the overall system. Hence, we have selected Support Vector Machine (SVM), Recurrent Neural Network Long Short-Term Memory (RNNLSTM), Bidirectional Encoder Representations from Transformer (BERT) and DistilBERT which a distilled version of the BERT, at different layer of the system as shown in Figure10.

The following subsections describe functionality of entire system along with details of each layer along with dataset used for each layer.

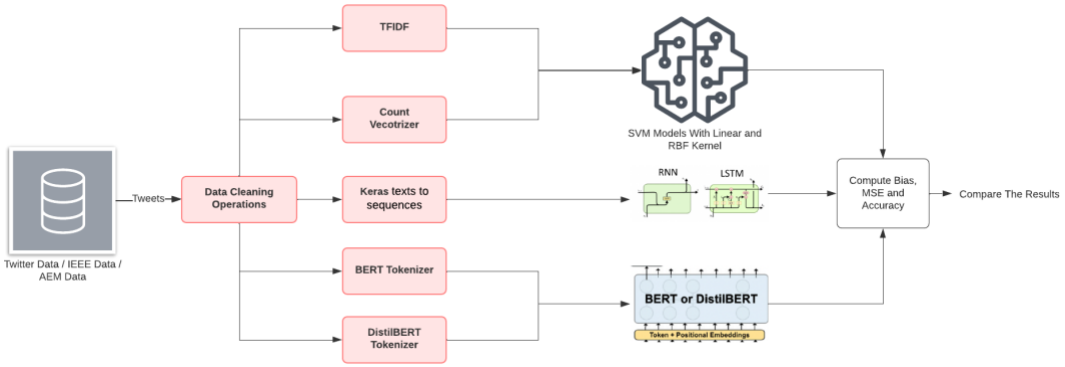


Figure 10: Dataflow and Architecture of Entire Process.

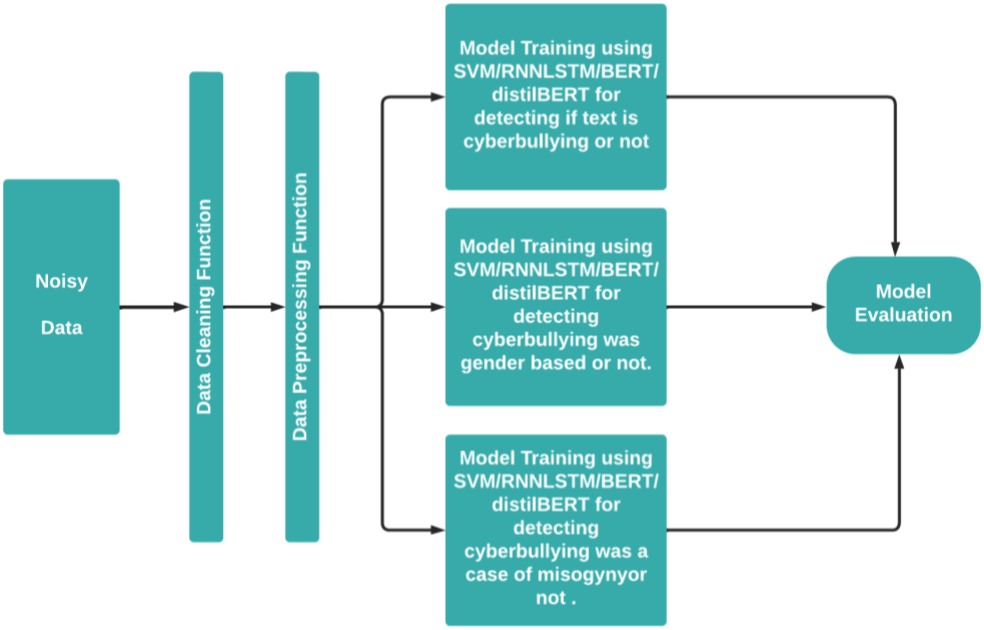


Figure 11: Functionality of Each Layer of Cyberbullying Detection Process.

3.3.1 Data Collection and Preprocessing

As described earlier and as shown in the Figure 11, our cyberbullying detection system consists of three layers, and each layer has their own functionality and thus we collected dataset for each layer separately so, that data is fully related to the functionality of the layer for which it is being used.

3.3.1.1 Data Cleaning

This is a very important step in the entire process, since the text collected from social media platforms contains various symbols like hash tag, and punctuations etc. So, we have created a common data cleaning operation using Natural Language Tool Kit (NLTK) library [30]. In the first step of the data cleaning, we have removed all the symbols and non-alphabetic contents and punctuation. Further for the next step we removed all the short words which has less than three characters as these words are mainly for grammatical requirements and does not contain the information for the basis of model to decide the class

of the text. So, we removed those kinds of words from the dataset. Further all these words are converted to vectors and will be used as the features. So, we must avoid the words which are not useful, because it will unnecessarily increase the feature vectors.

3.3.1.2 Data Preprocessing

After cleaning the data, we have removed all the unwanted and unrelated words, but there still exists redundancy in the dataset and it comes from the inflection in the words as several words get modified to align to the requirements of the grammar like tense and sometimes words get modified to better communicate about gender mood etc. Therefore, to reduce such redundancy we use the process of stemming. Stemming is the process of reducing the words to their root words by removing the suffixes from the words like "connecting" to "connect" after removing "ing" from the word. This helps in removing the inflection from the words and thus it helps in reducing the redundancy in the data.

After applying stemming on the data, we needed it to convert into some numerical forms, because the Machine Learning model will not work with the text directly. To have the best combination of word vectorizers and model variation in terms of kernel, we have used TFIDF (Term Frequency–Inverse Document Frequency) and Count Vectorizer methods to generate the feature vectors to send it to model for training. Following subsection explains how the TFIDF and Count Vectorizer works.

3.3.1.3 TFIDF

It is a very efficient method to perform vectorization process, this is a multiplication of two metrices, Terms Frequency and Inverse Document Frequency [31]. The term frequency is calculated by dividing the number of times a term is present in a document by total number of terms in that document, and inverse document frequency is calculated by

calculating the log of number of documents divided by number of documents that contain the term. Term Frequency and Inverse Document Frequency are calculated as shown in the equations 2 and 3 respectively.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}} \quad \text{.....(2)}$$

$$idf\ (\omega) = \log \frac{N}{dft} \quad \text{.....(3)}$$

There are several advantages of using the TFIDF. One major advantage is that it reduces the value for the words, which are present in all the documents and give more importance to the words which appears rarely in the documents. To understand the importance of TFIDF let's take an example of the two texts from two classes Cyberbullying and Non-Cyberbullying and if any word is present in both the text then a simple count based vectorizer may give equal importance to words in both classes and that will affect the accuracy of the model as we are providing with contradicting features. On the other hand, the TFIDF handle it with a tricky mathematical operation, taking the same case where we have words common in all the documents for both documents the term frequency is as usual now for inverse document frequency the result of division of number of documents and number of documents containing the particular term will be closer to 1 as the number of documents with the term will be closer to the total number of the documents, and log of the number closer to 1 will be near to 0. Similarly, for the word which are present in fewer number of documents will have higher value of TFIDF because the value of log of number of documents divided by number of documents with the term will be higher. Therefore, using this method, TFIDF helps in handling these scenarios very effectively.

3.3.1.4 Count Vectorizer

This subsection explains the second vectorizer, count vectorizer, we have used in our approach. The second vectorizer is a simple but very effective technique, where the words are converted to vectors based on the frequency of the words i.e., the number of times a word is present in a particular documents or text [32]. This seems a simple method, but it is one of the most common methods because in various application it has shown that the frequency of particular words does contains some useful information about various aspects of the text such as sentiment polarity.

3.3.2 Algorithm Used

As shown in Figure 10, we have used Support Vector Machines (SVM), RNNLSTN and highly pretrained BERT based models in our cyberbullying detection. These algorithms are explained in the following subsections.

3.3.2.1 Support Vector Machine

A Support Vector Machine is a supervised machine learning algorithm. It is a discriminating classifier that works by plotting data in a N-Dimensional space and then find the best suitable hyper-plane which can classify data distinctly into their respective classes and for finding the hyper-plane it uses concept of maximum margin hyper-plane. It can be used for both classification and regression, but it's mostly used for classification problems.

The main idea on which SVM works is that it tries to find the classifier or decision boundary such that the distance between decision boundary to the nearest data points of each classes is maximum (such hyper-planes are also known as maximum margin hyper-plane). That's why it's also known as maximum margin classifier.

Margin - A margin can be defined as the distance of the closest points to the decision surface. We can also say that the margin is the distance between the decision boundary and each of the classes. So, in Figure 12 we can see that points (X_1^1, X_2^1) and (X_1^2, X_2^2) are closest to the decision surface hence the distance between these point and decision surface is the Margin. Let's plot above points in the graph below to visualize the concept in more details-

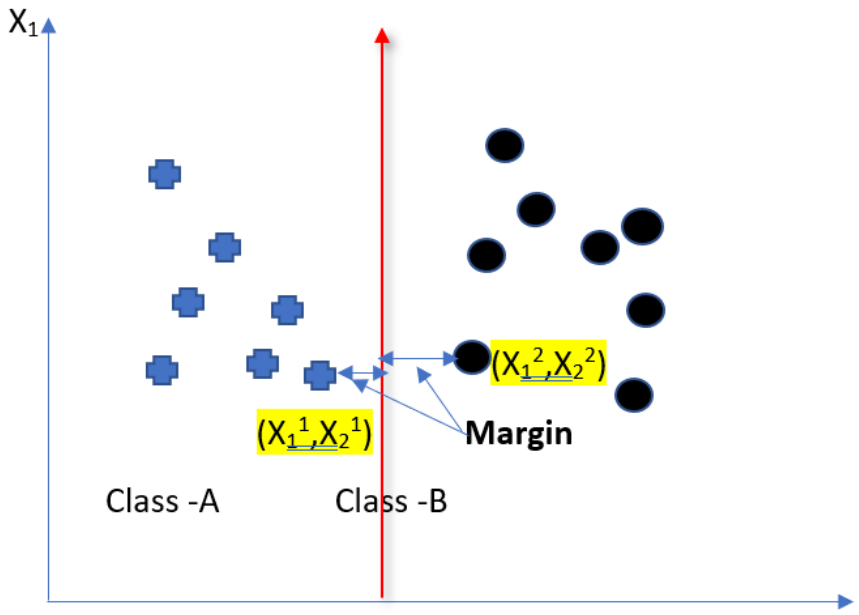


Figure 12: Concept of the Margin.

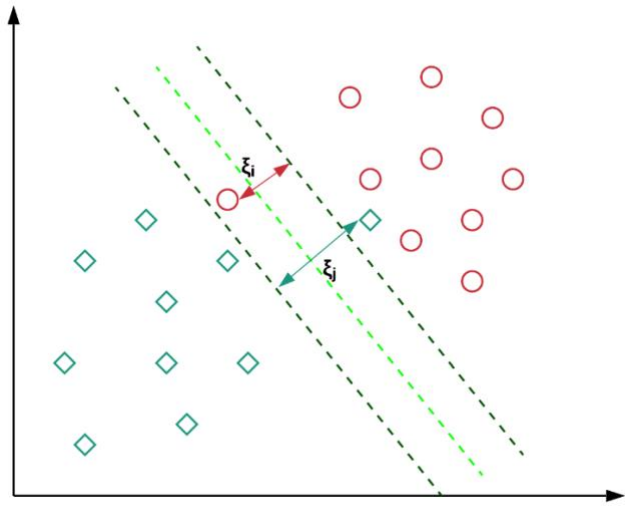


Figure 13: Concept of the Hyperplane.

The Figure 13 visualizes the concept of the margin and how margin is calculated using the closets points. The points which are near to the hyper plane are known as support vectors and these points are used to maximize the margin. Any change like deletion of these support vectors effects the orientation of the hyperplane. To classify our data into two classes cyberbullying and non-cyberbullying classes first task is to plot the word vectors from both classes in a N-dimensional space, then SVM will find a hyperplane which will maximize the margin between both the classes.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots(4)$$

3.3.2.2 RNNLSTM

In the field of natural language processing in particular, recurrent neural networks (RNN) and long short-term memory networks (LSTM) networks are particularly successful algorithms [41]. The RNN has an internal memory, a core feature that it picks up from

previous and current computations, and it excels at handling sequential data like text. Now, RNNLSTM is a redesigned RNN network that further simplifies the memory component. Input, output, and forget gates are the three crucial gates that it has. The forget gate is crucial because it uses the current input and the prior hidden state to evaluate the input, returning a 1 if the input should be kept and a 0 if it should be forgotten. This is one the most important part of the LSTM architecture.

3.3.2.3 Model Training with BERT based models

In addition to SVM and RNNLSTM, we use BERT based models for evaluation purposes. The concept and methodology of the BERT is explained as follows.

The BERT is a transformer-based machine learning technique for Natural Language Processing, which was pre-trained and developed by Google. BERT approach is based on attention-based mechanism, which helps the model to select the relevant context of a given word. It encodes the data in very useful manner, and it reads the text from both the directions and thus allows algorithm to have better understanding of the text. It first randomly masks the words in a sentence and then it tries to predict them, to predict the words it reads the text from left to right and right to left i.e., it uses full context of the word to predict them.

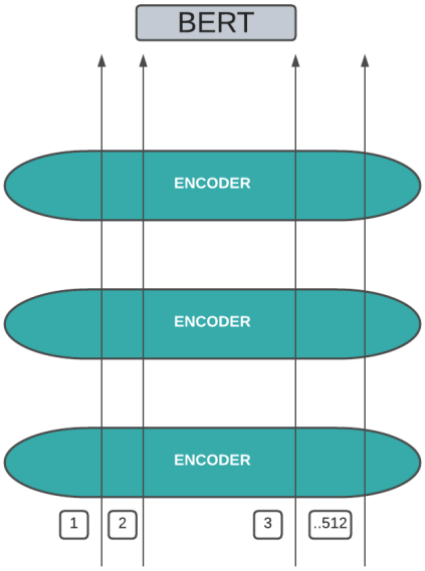


Figure 14: Functional diagram of BERT.

3.3.3 Layer - 1 – Detection of Cyberbullying

This is the first layer of our cyberbullying detection system. This layer is very important as this acts as an entry point to our system, as this layer detects that a given text is a case of cyberbullying or not. We collected a suitable dataset for this layer and then applied this dataset with different algorithms to see how different architecture performs in terms of Bias and Mean Squared Error.

3.3.3.1 Dataset

For this layer we have selected a twitter data, because it has variety of texts and from different regions and thus, we can say that the data is fully generalized and have sufficient variations in it. Moreover, we have ensured that data is balanced from each class. We took 5347 data points which belong to cyberbullying class and 5400 tweets from non-cyberbullying class. Balanced dataset is used to avoid any effect on Bias because of imbalanced dataset.

3.3.3.2 Training Models

We have performed the model training using several algorithms after doing the data cleaning and data preprocessing, for the first layer and tried to arrive at the best model architecture.

3.3.3.2.1 Training SVM models

The first algorithm we used is the Support Vector Machine with linear kernel and word vectors generated from the TFIDF approach. Then we tried and continued the experiment with the other kernel such as Radial Basis Function (RBF) and along with the vectors from the TFIDF vectorizer as features for our model. Finally, we tested the output vectors from the count vectorizer and linear and RBF kernel. We selected a final model as SVM with linear kernel and TFIDF vectorizer, since this combination performed well and gave best output when compared to other combinations.

3.3.3.2.2 Training RNNLSTM model

To evaluate our model architecture, we also performed experiments with RNNLSTM. Towards that we have used the cleaned text from the same method which we have used for other models and passed it to Keras text to sequence converter and trained the model and evaluated the performance accordingly.

3.3.3.2.3 Training BERT and DIStilBERT model

Another algorithm that we used is BERT. For training the BERT model, we have used the BERT tokenizer and as the first layer we have used the distilBERT tokenizer for distilBERT model. We have used the cleaned data here not the stemmed data, and then passed the output to BERT and distilBERT models to retrain to with our data. Then we

evaluated the model based on different parameters like accuracy, bias, mean square error, Recall and precision.

3.3.4 Layer - 2 – Gender-Based Cyberbullying Detection

This is the second layer in our overall cyberbullying detection system. Here, we are trying to find if the bullying was done based on the gender or not. This model once trained will get the input from layer 1 whenever the first layer detects a text as cyberbullying. Further the model will predict 1 for the case where the subject of the bullying is gender and 0 when subject of the bullying is not gender. Here, it needs to be noted that output 0 at this layer does not mean that the text is not cyberbullying-related, it just means that the subject is other than gender.

3.3.4.1 Dataset

For this layer we have used IEEE dataset, where the data is fully cleaned and labelled. Here, the dataset is divided into 5 classes: age, ethnicity, gender, religion, and text which include mainly harassment messages labelled the data as 0 for all classes other than gender-based cyberbullying and labelled it 1 for data belonging to gender-based cyberbullying class.

3.3.4.2 Training Models

Once we have the data ready for modelling, now for this layer too we will train multiple models for SVM, with different combinations of kernels and vectorizer techniques, RNNLSTM model and BERT model to further experiment with this layer.

3.3.4.2.1 Training SVM model

For the second layer, we performed experiment with SVM model. Here, we trained the model first with linear kernel and TFIDF data, then we trained SVM model with RBF

kernel and with TFIDF vectorizer output. We continued similar experiments with count vectorizer vectors and trained first model with linear kernel and then with RBF kernel. Then we evaluated the performance of each model to find the best SVM model and word vectorizer combination for the second layer.

3.3.4.2.2 Training RNNLSTM model

For the second layer also we performed experiment with RNNLSTM. Similar to first layer, text cleaning and preprocessing were performed and the model got trained. Finally, we evaluated the performance of the models.

3.3.4.2.3 Training BERT and DIStilBERT model

Similar to the first layer, we cleaned the text to train the models. For BERT model, we used BERT tokenizer and for distilBERT used distilBERT tokenizer and masking techniques. Then trained the BERT and distilBERT models and analyzed the outcomes.

3.3.5 Layer – 3 – Misogyny-Based Cyberbullying Detection

In the third layer, we created an algorithm that can detect if the cyberbullying is targeting a woman. This layer is the last layer, where we are trying to detect or extract some different information. The final combined layer is just a combination of models from each layer to make these layers to work as a single entity. Here we are trying to build a binary classifier, where we have labels 0 and 1, with 0 indicating non-misogyny and 1 indicating misogyny. We have again performed experiments with the SVM models including combinations of the different kernel and word vectorizer techniques. In addition, we have also performed the training with BERT model to see if we can build a more efficient model which will perform better in terms all the model evaluation techniques.

3.3.5.1 Dataset

For this layer, we selected Automatic Misogyny Identification (AMI) dataset. One important feature of this AMI dataset is that it contains data from the two languages Spanish and English, so for this layer we have first converted data in Spanish language to English language using GoogleTranslator. Then merged the data from these two languages in a single collection. The data contains total 6354 records. We applied the same data cleaning operations, as explained earlier in this section, to remove the stop words punctuations, symbols and other words, which does not have information that can help us to find if the cyberbullying is associated with misogyny or not. We applied the similar preprocessing steps for both SVM and BERT models which we used for layers 1 and 2 i.e., stemming with TFIDF and CountVectorizer for SVM based models, keras text to sequence converter for RNNLSTM and BERT tokenizer for BERT.

3.3.5.2 Training Models

Once we have the data ready for model, in this layer too we trained multiple models for SVM with different combinations of kernels and vectorizer techniques. Then we trained SVM, RNNLSTM, BERT and distilBERT models.

3.3.5.2.1 Training SVM model

For this layer too we trained SVM models with the combination of different kernel: linear and RBF and word vectorizer techniques: TFIDF and Count Vectorizer. Then we evaluated the results to find the best SVM layer for this layer.

3.3.5.2.2 Training RNNLSTM model

Similar to the first two layers, we performed text cleaning and preprocessing and then we trained the model. Then we evaluated the performance of the model.

3.3.5.2.3 Training BERT and distilBERT model

Similar to the first two layers, we cleaned text and performed text preprocessing techniques to convert data in suitable forms and trained the BERT and distilBERT models. Finally, we calculated different efficiency measurements techniques to analyze the performance of each model.

3.3.6 Combined Layer

In the combined layer level, we did not perform any model training, since we just stacked models from each layer and then evaluated the performance of the entire system. Here, we selected best model from each layer such as for the SVM we selected the best performing model from each layer as SVM model with linear kernel and TFIDF vectorizer for first layer, SVM model trained with linear kernel and count vectorizer for second layer and finally SVM kernel trained with RBF kernel and count vectorizer vectors for third layer and stacked them to form one system and passed test data and then analyzed the performance of overall system.

Similarly, for RNNLSTM, we collected RNNLSTM model from each layer and stacked them together and passed test data from the system and evaluated the outcomes. Then we applied the same process for BERT and distilBERT models from each layer and stacked them together. In this combined layer experimentation, the data first passes from the model from first layer and if the text is detected as cyberbullying, then the text gets passed to the second layer to detect if the cyberbullying is gender-based or not. Only if the text is gender-based cyberbullying, then the text gets passed to next layer. In the final layer, the model detects if the cyberbullying is of misogyny class or not.

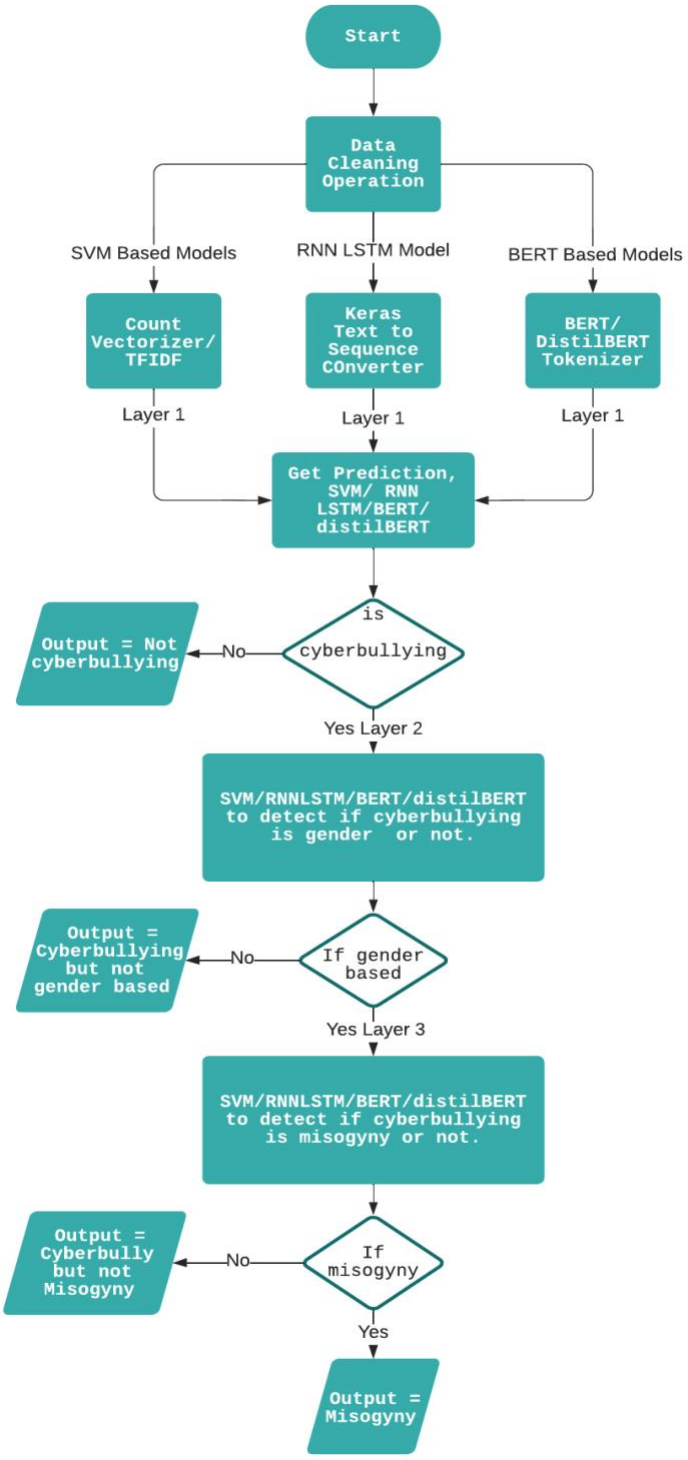


Figure 15: Dataflow Diagram Showing Functionality of Combined Layer.

3.4 Model Evaluations and Discussion of Results

3.4.1 Model Training

This section describes the model evaluation and results for SVM, RNNLSTM, BERT, and DistilBERT models for all layers described in section 3.

3.4.2 Model Evaluation for Layer 1

3.4.2.1 SVM with linear kernel and TFIDF vectors

We first performed the experiment with the Support Vector Machines and evaluated four models using SVMs. For the first model we have taken word vectors generated from TFIDF and divided it into training and testing sets, 70% of the data were used for training and remaining 30% data were used for testing the model. Also, for the first model we used SVM with the linear kernel. Results for SVM with linear kernel are shown in Table 4.

Model and Vectorizer Technique	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
Results with SVM with Linear Kernel and TFIDF	80.4341%	84.8729%	77.9613%	81.2704%	25.2%	19.57%
Results with SVM with RBF Kernel and TFIDF	80.186%	84.3769%	77.8604%	80.9878%	25.18%	19.81%
Results with SVM with RBF Kernel and Count Vectorizer	80.201%	85.4031%	78.0358%	81.5534%	25.22%	19.8%
Results with SVM with Linear Kernel and Count Vectorizer	79.34%	83.95%	77.58%	80.6418%	25.16%	20.66%

Table 4: Results of SVM Models with TFIDF and Count Vectorizer.

To evaluate the trained model, we used the testing set and predicted the data from the model and then using true labels and predicted labels, the model gave 80.43% of accuracy which is very encouraging. In addition, precision and recall were used for evaluation. If model has higher is precision, it means it is more efficient in picking the more relevant data and higher recall means that algorithm returns most of the relevant data. In our evaluation, we got a precision of 77.96% and it means that 77.96% of data which has been classified to a particular class belongs to that class and the recall is 84.87%, which indicates that around 84% of data was correctly labeled by the model.

In addition, we also calculated the F1 score, and it's the harmonic mean of precision and recall the F1 score combines the precision and recall, and we can see here that we got a F1 score of 81.27%, which indicates that model has very descent precision and recall.

To further evaluate the model, we have also calculated Bias and Means Square error, which is considered as the systematic error occurred in the model due to incorrect assumptions. So, if a model has higher bias, then it means that model is not capable of capturing the trends and real information in the dataset and it will result in higher error rate. For the model here we obtained a bias value of 25.2%. While the value is not too high, it also says that model has some incorrectness in it and also this is also getting reflected in mean square error value of 19.57%.

The equations for the metrics used in the model evaluation are as follows:

Precision = (True Positive)/ (True Positive + False Positive)..…… (5)

Recall = (True Positive)/(True Positive + False Negative) ……..(6)

F1 score = 2*(True Positive)/ (True Positive + False Negative) …….(7)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad \text{.....(8)}$$

The results in the Table 4 give a fair idea of how much the model is efficient in terms of fairness, the bias is not too low but can be considered as decent value. On top of that the model has a good accuracy, precision, and recall and thus we can say that model has performed decently in terms of all the parameters of the fairness.

3.4.2.2 SVM with RBF kernel and TFIDF vectors

We further wanted to improve the model efficiency and as explained in section 3, we trained a SVM model with RBF kernel and by taking vectors generated from TFIDF method. The results of this experiment are shown in Table 4.

As shown in Table 4, we can see there is a slight decrease in the accuracy, recall, precision and a slight decrease in the bias and MSE. This shows that although the model performance has not decreased a lot, but it does not show much improvement. Also, there is a slight decrease in bias but at the same time mean square error has increased, so when we consider all the evaluation metrices we can say that it is less fair than the linear kernel.

3.4.2.3 SVM with linear kernel and count vectorizer vectors

For the next experiment we used the word vectors generated from the count vectorizer and trained the SVM model with linear kernel. The results are shown in Table 4.

As shown in Table 4, the results indicate that there is a slight decrease in the accuracy, Recall, F1 score, Precision and also, we can see that there is an increase in the Bias and mean square error. Therefore, we can say that the model is much lower in terms of fairness than the model trained with TFIDF and linear kernel approach.

3.4.2.4 SVM with RBF kernel and count vectorizer vectors

For the next experiment, we used the word vectors generated from the count vectorizer and trained the SVM model with RBF kernel. The results are shown in Table 4.

As indicated in Table 4, the results demonstrate that we have a slight increase in the accuracy, Recall, and Precision. Also, at the same time there is an increase in the Bias. Moreover, mean square error is same as the model with linear kernel and TFIDF vectorizer. So, we can say that the model is almost equal in terms of fairness with models trained with TFIDF and linear kernel.

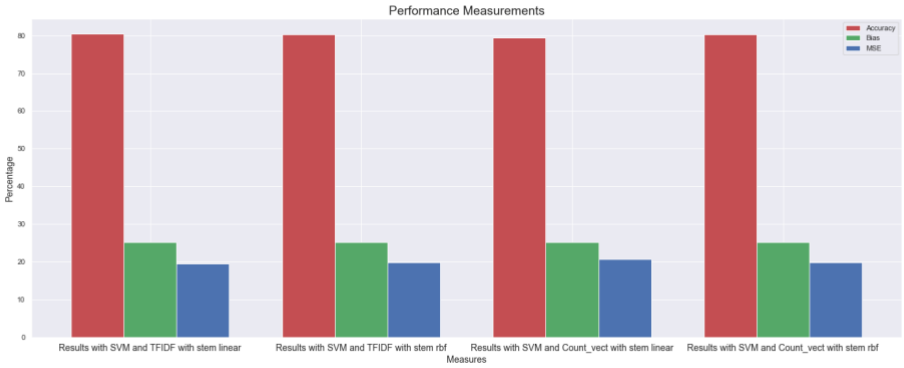


Figure 16: Comparison of SVM Models in Layer 1 Cyberbullying Detection.

3.4.2.5 Model evaluation for RNNLSTM

For the first layer, we trained the RNNLSTM model and found that the model has not performed well, and the accuracy was lower than SVM as well as BERT and distilBERT models. As shown in the Table 5, the model gave an accuracy of 78.64%, with 79.47% recall, 77.94% precision, the F1 score was 78.7 %. The Bias was nearly identical to the SVM model at 25.01%, but there was an increase in the mean square error at 21%.

3.4.2.6 Model evaluation for BERT

As shown in Table 5, the fine-tuned BERT model performed much better than SVM and RNNLSTM models and gave 90% accuracy and good recall of 88.88%, precision(90.72%)and F1 score was at 89.79%. However, bias was almost similar to SVM and RNNLSTM model at 25% and the mean square error got reduced to 10%.

3.4.2.7 Model evaluation for distilBERT

The distilBERT model evaluation produce similar values as BERT model. As shown in Table 5, distilBERT model resulted in an accuracy of 89%. However, the bias and mean square error were almost equal to BERT, which were at 25% and 10% respectively.

Model Type	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
SVM with SVM with Linear Kernel and TFIDF Vectorizer	80.4341%	84.8729%	77.9613%	81.2704%	25.2%	19.57%
Results with RNNLSTM	78.6412%	79.4752%	77.9412%	78.7007%	25.01%	21.36%
Results with BERT	90%	88.8889%	90.7216%	89.7959%	25.01%	10%
Results with DistiltBERT	89.4%	91.7355%	87.0588%	89.336%	25.04%	10.6%

Table 5: Comparison of Results from Different Models in Layer 1 Cyberbullying Detection.

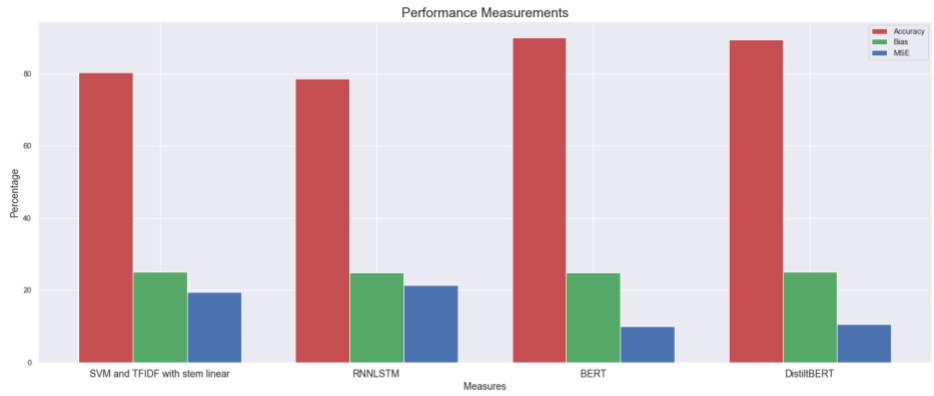


Figure 17: Comparison of SVM, RNNLSTM, BERT and Distilbert Model in Layer 1 Cyberbullying Detection.

3.4.3 Model evaluation for layer-2

In this section, we explain the evaluation of different models used in level 2 of our cyberbullying detection method using supervised learning techniques.

3.4.3.1 SVM with linear kernel and TFIDF vectors

For second layer, the SVM model with linear kernel and vectors from TFIDF technique produced promising results. As shown in the Table 6, accuracy of the model achieved for is at 95.38%, recall was at 98.39%, precision was at 95.91%, also the F1 score was at 97.45%. The bias and MSE were much lower at 15%and 4% respectively. So, overall performance of the model was much better.

3.4.3.2 SVM with RBF kernel and TFIDF vectors

The SVM trained with RBF kernel and TFIDF vectors performed similar to the linear kernel-based model and as shown in the Table 6, the model resulted in an accuracy of 95.41%, 98.48% recall, 95.86% precision, 97.15% F1 score. The model also had bias of 16% and 4.5% of MSE. The performance of this model is almost similar to the kernel-based model.

3.4.3.3 SVM with linear kernel and count vectorizer vectors

The experiment of linear kernel with count vectorizer vectors, SVM model performed similar to earlier two models and resulted in 95.88% accuracy, 97.85% recall, 97.06% precision, 97.45% F1 score, as well as bias of 15% and 4.12% MSE.

3.4.3.4 SVM with RBF kernel and count vectorizer vectors

For the last combination of the SVM with RBF kernel and count vectorizer, the model performed very closely with the other SVM models of this layer and produced 95.77% accuracy, 97.85% recall, 96.39% precision, which is slightly lower than last layer, 97.40% F1 score, as well as bias of 15% and 4.23% MSE. These values suggest that the model performed similar to other SVM model for this layer.

Model and Vectorizer Technique	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
Results with SVM with Linear Kernel and TFIDF Vectorizer	95.3878%	98.3992%	95.9142%	97.1408%	16.27%	4.61%
Results with SVM with RBF Kernel and TFIDF Vectorizer	95.413%	98.4834%	95.8684%	97.1583%	16.27%	4.59%
Results with SVM with Linear Kernel and Count Vectorizer	95.88%	97.85%	97.06%	97.4557%	15.67%	4.12%
Results with SVM with RBF Kernel and Count Vectorizer	95.7734%	98.4385%	96.3914%	97.4042%	15.69%	4.23%

Table 6: Results of SVM Models at Layer 2 of Cyberbullying Detection.

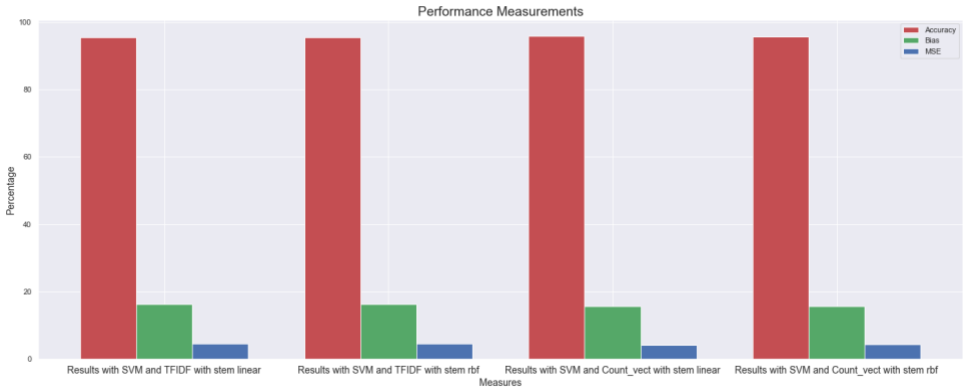


Figure 18: Comparison of SVM Models for Layer 2 of Cyberbullying Detection.

3.4.3.5 Model evaluation for RNNLSTM

In second layer also we performed the training of RNNLSTM model. The RNNLSM model performed almost similar to the SVM models in terms of all performance measurement parameters that we were considered. The model achieved 95.44% accuracy, further it achieved 98.07% recall, 96.32% precision, and a F1 score was around 97.2 %, the Bias was nearly to the SVM model at 15% but there was a minor increase in the mean square error at 4.55% .

3.4.3.6 Model evaluation for BERT

The training of BERT model performed much better than SVM and RNNLSTM models and achieved a perfect 100% accuracy, recall, and precision. F1 score resulted at 89.79%. In addition, we got 0% bias and mean square error.

3.4.3.7 Model evaluation for distilBERT

The distilBERT model too produced a very good results and gave an accuracy of 98%, with 98% recall, 99.35% precision and 98.72% F1 score. However, there was a slight in increase in bias at 16% and mean square was at a low level of 2%.

Model Type	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
Results with SVM with Linear Kernel and Count Vectorizer	95.88%	97.85%	97.06%	97.4557%	15.67%	4.12%
Results with RNNLSTM	95.4471%	98.0676%	96.3272%	97.1896%	15.86%	4.55%
Results with BERT	100%	100%	100%	100%	0%	0%
Results with DistiltBERT	98%	98.0892%	99.3548%	98.7179%	16.89%	2%

Table 7: Comparison of SVM, RNNLSTM, BERT and distilBERT Models in Layer 2 Cyberbullying Detection.

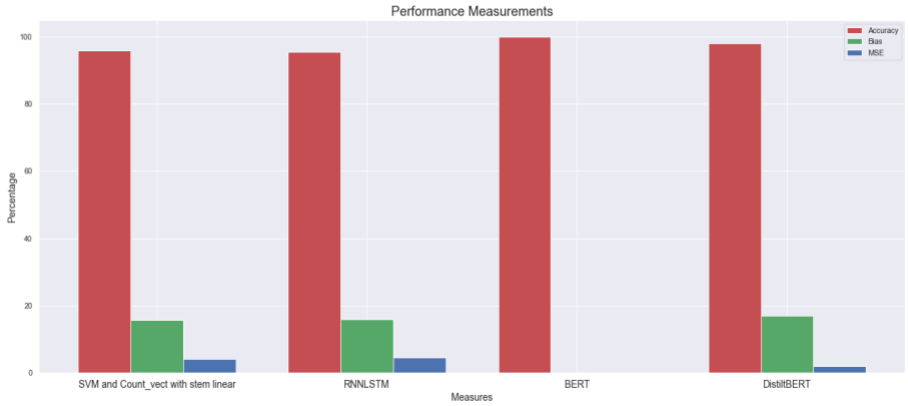


Figure 19: Comparison of SVM, RNNLSTM, BERT and Distilbert Models in Layer 2 Cyberbullying Detection.

3.4.4 Model evaluation for layer-3 Cyberbullying Detection

For third and the final layer of our cyberbullying detection framework, we performed similar training on the different models like other two layers.

3.4.4.1 SVM with linear kernel and TFIDF vectors

In the third layer, the SVM model with linear kernel and vectors from TFIDF vectors gave some decent results. As shown in Table 8, the accuracy was at 75.81%, recall was at around 77.96%, with this precision was at 75.41%, the F1 score was at 76.7%. Also, as shown in Table 8, the bias and MSE were higher than layer 2 layer at 25% and 24% respectively.

3.4.4.2 SVM with RBF kernel and TFIDF vectors

In Layer 3 of our framework, as shown in Table 8, SVM model with RBF kernel and TFIDF vectors gave around 76.73% accuracy, recall of around 80.16%, with 75.63%

precision, and 77.83% F1 score. Also, bias was at around 25% and 23.27% MSE As shown in the Table 8, the performance of this model is almost equal to the kernel-based model.

3.4.4.3 SVM with linear kernel and count vectorizer vectors

For SVM model with count vectorizer output, and linear kernel, the performance of the model came down to 72.68% accuracy, 73.75% recall, 73.05% precision, 73.39% F1 score, 25% bias and with MSE increased to 27%.

3.4.4.4 SVM with RBF kernel and count vectorizer vectors

For RBF kernel with count vectorizer output, the SVM model performed slightly better than other SVM models for this layer and as shown in the Table 8 produced 76.89% accuracy, 84.61% recall, 73.93 % precision and, 78.91% F1 score. Also, as shown in Table 8, the bias was at 25.53% and MSE too was slightly lower than earlier models at 23.11%.

Model and Vectorizer Technique	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
Results with SVM with Linear Kernel and TFIDF Vectorizer	75.813%	77.9661%	75.4098%	76.6667%	25.02%	24.19%
Results with SVM with RBF Kernel and TFIDF Vectorizer	76.7276%	80.1595%	75.635%	77.8316%	25.08%	23.27%
Results with SVM with Linear Kernel and Count Vectorizer	72.68%	73.75%	73.05%	73.3967%	24.99%	27.32%
Results with SVM with RBF Kernel and Count Vectorizer	76.8902%	84.6062%	73.9312%	78.9093%	25.53%	23.11%

Table 8: Results of SVM Models at Layer 3 Cyberbullying Detection.

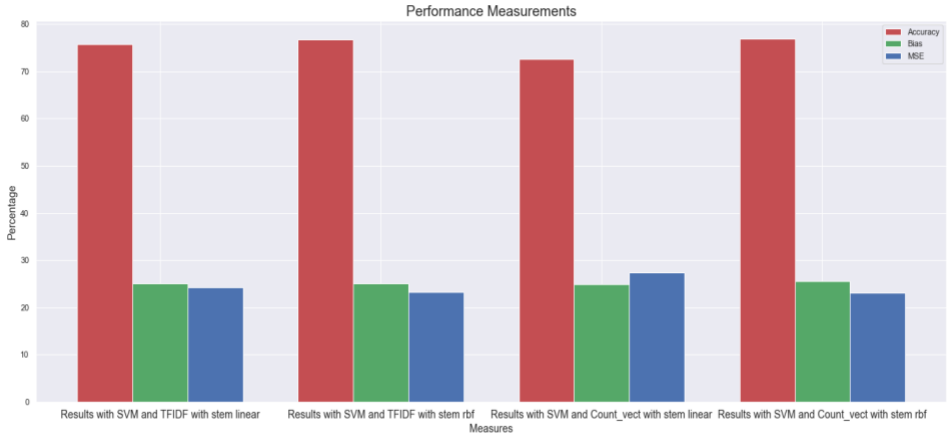


Figure 20: Comparison of SVM Models for Layer 3 Cyberbullying Detection.

3.4.4.5 Model evaluation for RNNLSTM

In third layer we also performed the training of RNNLSTM model. The RNNLSM model performed almost similar to the SVM models in terms of all metrics that we were considered. The model achieved 72.85% accuracy. Further, it achieved 74.98% recall, 73.74% precision, and F1 score around 74.34 %. The Bias was slightly higher from the SVM model at 27.14%, but there was a minor decrease in the mean squared error at 27.95% .

3.4.4.6 Model evaluation for BERT

The training of BERT model with data from the third layer performed similar to the SVM and RNNLSTM models and as shown in the Table 9, the BERT model resulted in 75.5% accuracy, 79% recall, 73.83% precision and 76.33% F1 score with almost equal bias of 25% and SME of 24%.

3.4.4.7 Model evaluation for distilBERT

The distilBERT model for this layer performed better than all other algorithms and gave better results with an 88.4% accuracy, 85.2% recall, 93.28% precision and 89.056% F1 score. While bias was at similar level of 25%, SME was lower than other models with a value of 11.6%.

Model Type	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
Results with SVM with RBF Kernel and Count Vectorizer	76.8902%	84.6062%	73.9312%	78.9093%	25.53%	23.11%
Results with RNNLSTM	72.856%	74.9625%	73.7463%	74.3494%	24.95%	27.14%
Results with BERT	75.5%	79%	73.8318%	76.3285%	25.12%	24.5%
Results with DistiltBERT	88.4%	85.1986%	93.2806%	89.0566%	24.94%	11.6%

Table 9: Comparison of SVM, RNNLSTM, BERT and distilBERT Models at Layer 3 of Cyberbullying Detection.

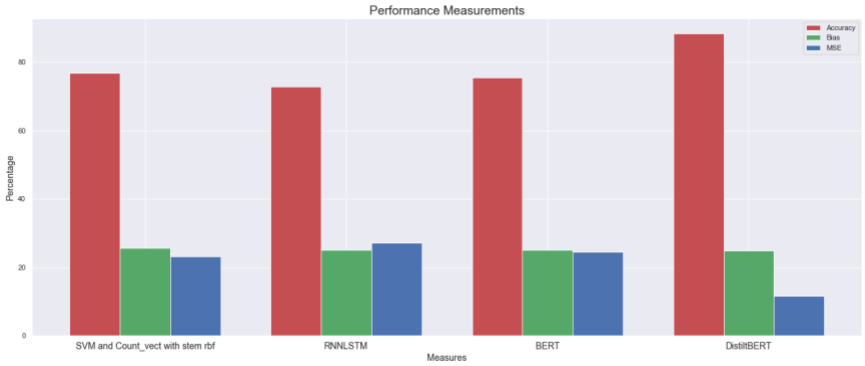


Figure 21: Comparison of SVM, RNNLSTM, BERT and Distilbert Models at Layer 3 of Cyberbullying Detection.

3.4.5 Model evaluation for combined layer

As described in section 3, the combined layer is different than other three other layers. Here we stacked models from each layer of same underlying algorithm and tested its performance based on accuracy, recall, precision, F1 score, bias and mean square error.

3.4.5.1 Combined layer with SVM

The combined layer with best SVM models from each layer (layer1,2 and 3) gave average results with 43.5% accuracy, good recall of 84.62%, but poor precision of 29.53%. The bias was a bit higher at 43% and mean square error was also higher at 56.5%.

3.4.5.2 Combined layer with RNNLSTM

As shown in Table 10, the combined RNNLSTM models from each layer resulted in slightly better results with 56% accuracy, good recall of 92.30%, but average precision score of 36.36%. However, the bias was lower at 35% and mean square error was also lower at 44%.

3.4.5.3 Combined layer with BERT

As shown in Table 10, the BERT model in the combined layer resulted in 48.0% accuracy, good recall of around 92.11%, poor precision of 31.82%, and the average F1 score of 47.3%. However, the bias and mean square error was higher at 41% and 52% respectively.

3.4.5.4 Combined layer with distiltBERT

distiltBERT model performed better than all other models and gave decent results with 65.5% accuracy, decent recall of 78.85%, average precision score of 41.41%. However, the bias and mean square error was also lower than other models at 24% and 34.5% respectively.

Model Type	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
Results with SVM	43.5%	84.6154%	29.5302%	43.7811%	42.76%	56.5%
Results with RNNLSTM	56%	92.3077%	36.3636%	52.1739%	35.24%	44%
Results with BERT	48.0%	92.1053%	31.8182%	42.2973%	41.96%	52.0%
Results with DistiltBERT	65.5%	78.8462%	41.4141%	54.3046%	24.76%	34.5%

Table 10: Comparison of SVM, RNNLSTM, BERT and distilBERT Models at Combined Layer of Cyberbullying

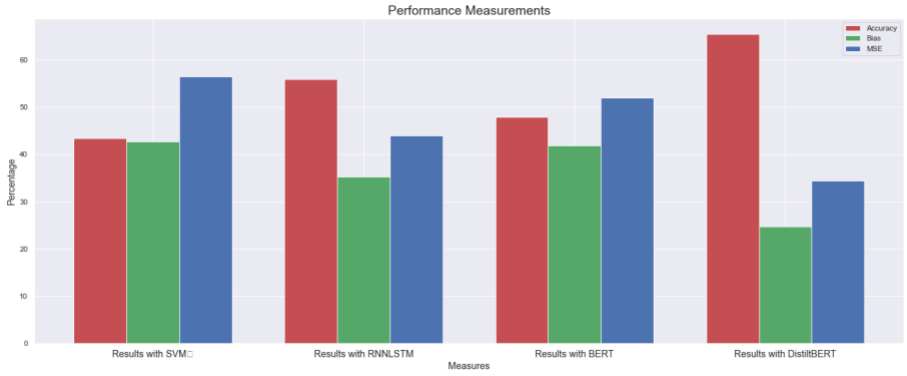


Figure 22: Comparison of SVM, RNNLSTM, BERT and distilBERT Models at Combined Layer of Cyberbullying Detection .

As shown in the Table 10 and Figure 22, the important aspect of the combined approach is that, when model was working independently then it gave some decent results, but with combined approach, the efficiency of the system came down. This due to the difference in the data sources in each layer, which can be resolved with more robust dataset.

3.5 Research Contribution

Through this work we have made several contributions, firstly we have performed experiments with different algorithms such SVM, RNNLSTM, BERT and distilBERT to showcase that using the more advanced models we build more effective algorithm which is very much efficient in terms of bias and fairness. Further in this work we have created models for three important aspects of the cyberbullying, the first model is to detect the cyberbullying, the second model detects the gender based cyberbullying and final layer detects if the bullying was a case of misogyny, this work will help to detect and prevent bullying as well as it will help to find more detailed analysis about cyberbullying such as number of cases of gender based bullying and cases of misogyny on social media. The work can be extended to further build models to detect other subjects of cyberbullying such as religion, ethnicity, age etc. Overall, this work has provided an efficient model to detect cyberbullying over social media platforms with decent bias and fairness.

3.6 Future Work Recommendations

As described in Section 4, the experimental results are promising. The proposed study and findings in this work has laid a very fruitful ground for future development and improvements. In future work, if all the layers are trained with more robust data with fully balanced classes and from similar background then the entire system can show a promising

result. Also, the layers can be tested with data from the different domains, where we need multilayer detection and classification mechanism.

3.7 Conclusion

Cyberbullying is one the most important issue in our society. It affects the victim to great extent and also it is equally affecting the person, when any system detects cyberbullying with bias. So, it's very important that we must put efforts to improve the fairness of the system's decision-making capabilities. In this work, we have presented few new scenarios to evaluate the Bias of the models and when we changed the underlying model to more advanced models, then we observed an improvement in the performance evaluation metrics such as accuracy, precision, recall, F1 score, bias and mean square error. If we can combine the algorithms-based outcomes with other efforts that focus on handling the Bias coming from the dataset, then we can further produce promising results and make the entire process more effective. Future work will be dedicated to this.

CHAPTER IV

PERFORMANCE EVALUATION OF CYBERBULLYING DETECTION ALGORITHMS FOR BIAS AND FAIRNESS

4.1 Introduction

We are living in the era of Artificial Intelligence, and we can see many implementations of machine learning around us. For example, ML models can analyze the comments and reviews of the customers to find even more in-depth details about user's feeling towards any product. ML-based models drive facial recognition system in our laptops and mobile phones. We also see many implementations of the machine learning in the medical domain [34]. As we see such growth in the use of the machine learning algorithm in critical domains, it is very crucial to verify the fairness of the machine learning algorithm. It is also important that research should be done to develop different methodology to improve the fairness of the machine learning algorithms. When we mention about fairness of the model, we also refer the bias, mean square error, precision, recall and F1 score with accuracy. These parameters are very important to analyze the performance of the model. Since, we cannot decide on the performance of the model with

accuracy alone. For example, if any model has very high accuracy but with higher bias then we cannot consider it as a fair model, and hence we cannot utilize it for critical domains. So, we considered all these critical performance measurement parameters in our evaluation experiments and then we have proposed a mechanism where we have shown that with change in the underlying model architecture, we can improve the performance of the model in terms of all the important performance measurement of the model including bias and fairness.

Further, in this work we have also focused on important methodology of machine learning which is weak supervision. In machine learning, the most important part is the data, which is the base through which the machine learning models learns.

The data has two major components: features and target value. The feature set extracts the information and then build a relationship with the target label. This mathematical relationship is used with unseen dataset, where we have only features. This mathematical model is used to predict the corresponding label. While, it seems to be a very easy implementation, however it has one limitation that we need to have fully labeled data. In this big data era, though we have plenty of data available for different domains, labeling such large amount of data is a time taking and costly work. Although, we can use unsupervised learning, it has its own limitations with respect to textual data. One main reason behind this is that each text has a different way of expressing similar information and utilizing such information effectively using unsupervised learning is not that much reliable. So, researchers have come up with the idea of weak supervision [35].

In this work, we have utilized the idea of weakly supervised learning, where we have showcased here that with weak supervision, we can build a very reliable and effective

model. The idea of weak supervision is based on weak labels and then we combine these labels to generate the final label. Here we have also considered the process of calculating the final labels. Here we have taken both the methods of calculating the final labels, which are most vote and averaged method. We have also implemented the averaged method in slightly different way, which has been explained in the methodology section.

Another important aspect of machine learning related research work is selecting a particular domain. Here we have selected cyberbullying domain, for the following reasons: (a) lately we have seen a huge increase in the cases of cyberbullying [2]. Further we see a surge in social media users, which is one of the main sources of cyberbullying cases. On the other hand, the social media has many advantages. For example, it has made people more informed, and it gives a very easily accessible platform for the people from any geographical region to share their opinion and thoughts on different issues, with that it has brought world more closer and well connected. Therefore, we have some responsibilities to make these platforms safer for everyone so that social media can be utilized fully for its good part. (b) The second reason to select cyberbullying domain is because the text from the cyberbullying cases is more complex and due to the case of negative sentiment. One important point here is that we cannot consider a negative texts or thoughts as bullying.

Rest of the chapter is organized as follows: Section 2 describes the related work of weak supervision, and related work with respect to fairness of the algorithms in natural language processing. In Section 3, we have provided a detailed explanation of the methodology which includes data preprocessing, generation of weak label, calculation of final labels and all the algorithms and model training process. In section 4, we have presented a detailed analysis of results in terms of fairness of models. Section 5 provides

comparison of results obtained using fully supervised learning and weakly supervised learning. Further, section 6 discusses about future works that can be performed in the various domains using weak supervision and algorithmic fairness. Finally, section 7 concludes our overall study.

4.2 Related Works

There are few works exist in the literature with respect to fairness and weak supervision. One such work is proposed for fair generative model with weak supervision. The work by Mehrabi [29], focuses on utilizing the weak supervision and prepare a fair generative model. Also, it has focused on one fact that the dataset might contain bias due to social and other impacts. One fact with this work is that it is a generative model but lays down a fact that we can build fair model, even with the bias in dataset. Our proposed work also considers this, but we have implemented it with cyberbullying use case. Another related work by Kai-Wei Chang et al. [38] presents a detailed study to deal with bias in the system. Our work can be considered as one of the approaches to handle fairness of the model and our work also includes the idea of weak supervision. Here as we know that bias comes from data too. Another related work which is focused to deal with the bias in dataset uses Causal Bayesian Networks (CBNs) [44].This work is limited to only dealing with bias in the dataset, but our work is mainly concerned about handling bias at algorithm level. So, if we can combine this work with our work then it can become a very useful implementation for addressing fairness and model bias issues.

Thus, there are few works done in the field of algorithmic fairness and weak supervision. However, as per our knowledge no related work exits in dealing with fairness of the model in weak supervision. Our work is focused on this aspect. We have not only

shown the bias in weak supervision, but we have also presented the idea to deal with it using some calculation changes in calculating the final label using weak labels. We have performed performance evaluation experiments and demonstrated that by using more advanced state of the art machine learning algorithms, bias can be reduced, and fairness can be increased.

4.3 Methodology

The most important part of this work is to conduct performance evaluation experiments for comparing bias and fairness with respect to weakly supervised learning and fully supervised learning algorithms. For that we have collected a large number of unlabeled data, and then trained multiple models such as commonly used Support Vector Machines to most advanced highly pre-trained models such as BERT and DistilBERT. We will go through each step in the details in subsequent section below.

The methodology for our experimental approach is explained as follows.

4.3.1 Datasets

The dataset is the most important part of any research in the field of Machine Learning. The biggest source of the cyberbullying is the social media and that is why we have collected our dataset from the social media. These are the collection of different texts generated by the users on different social media platforms such as Twitter, Youtube Comments, Facebook etc. One important aspect of the dataset collected from social media is that it contains both information and noises. Further, some social media platforms have the limitations on the number of the words that can be sent or posted. So, user try to express their thought or message in more concise way and as a result it becomes very tough to derive some decisive information that can help us to evaluate it to classify it for sentiments,

cyberbullying, crime news detection etc. On top of that the data we have is not having label, so it further possesses the challenge for the Machine Learning algorithms. We have collected the dataset from the sources such as IEEE [10] and Kaggle [43]. The IEEE dataset is a mix of texts from different cyberbullying subjects. It also has texts from non-cyberbullying texts, which we have collected the texts from each subject and merged it to form one collection of cyberbullying case. Kaggle dataset is a collection of text messages sourced from Twitter. We have collected the dataset and merged it in to one set for further process.

4.3.2 Data Preprocessing

For natural language processing use case, the data preprocessing is slightly different from other Machine Learning use cases, and it plays an important role in the extracting useful information. In our proposed work, the preprocessing of the data is a two-step process, the first step is data cleaning with stemming, and second step is to convert data into numbers.

4.3.2.1 Data Cleaning

The data which we have collected is from the social media platform and currently people apply several other tools apart from texts such as the use of symbols for example hash tags (#), and other symbols like "@", as well as using emojis with their messages. These symbols and emojis makes the messages interesting but when we see these in terms from Machine Learning perspective, these acts as noises or unnecessary characters using which we cannot extract information to find if the texts belong to cyberbullying or not and further these texts have punctuation which are again not useful for our use case. So, we have removed all these unnecessary emojis, symbols, and punctuation. To accomplish

these, we have used a very commonly used library Natural Language Toolkit (NLTK [30]. Further, in the natural language texts we have certain stop words like myself, we, our, ours, ourselves, you, you're, you've, you'll, you'd, your, yours, yourself and similar to the symbols and the punctuations these words are too not useful for our algorithm and thus using the NLTK library first we have collected these words using NLTK library and removed it from all the texts. To make sure that we have the texts with important words only, we have also made sure to remove short words which of one character. Now after this process we have data ready for further preprocessing.

4.3.2.2 Stemming

The dataset we have here are the texts and these texts are form natural language which we humans use for conveying our thoughts or message, so it contains same words in different formats due to grammatical requirements like working and worked are same words in different forms and there are several such words, and we should also consider one fact that each of these words will convert as an individual feature and thus if these words in different forms are there in our dataset then we will have high number of features and that will affect the training process. So, we need to bring down these words to the root words. The process of converting the inflated words to its words are called lemmatization, and for this process we have used Porter stemmer of NLTK library. One important aspect of the porter stemmer is that it chops of the affixes without considering if the resultant word is meaningful word or not. But still, it works in many scenarios.

4.3.2.3 Word to Vectors

The next step is to convert the words into numbers or vectors. The main reason behind is that the machine learning models are based on mathematical concepts. In order

to apply the math to the texts, we need to convert it into the numbers. To accomplish this, we have several methods. For common machine learning models, the two most used methods are TFIDF (Term Frequency and Inverse Document Frequency) and count vectorizer.

TFIDF (Term Frequency–Inverse Document Frequency) –The TFIDF is a very efficient method to perform vectorization process: this is a multiplication of two metrices, Terms Frequency and Inverse Document Frequency [31]. The term frequency is calculated by dividing the number of times a term is present in a document by total number of terms in that document, and inverse document frequency is calculated by calculating the log of number of documents divided by number of documents that contain the term. Term Frequency and Inverse Document Frequency are calculated as shown in the equations 9 and 10 respectively.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}} \text{.....(9)}$$

$$idf\ (\omega) = \log \frac{N}{dft} \text{ .....(10)}$$

There are several advantages of using the TFIDF. One major advantage is that it reduces the value for the words which are present in all the documents and give more importance to the word which appears rarely in the documents. A natural question that comes to us is why this functionality is important? To answer this, let's take an example of the two texts from two classes Cyberbullying and Non-Cyberbullying. If any word is present in both the classes, then a simple count based vectorizer may give equal importance to words in both classes and that will affect the accuracy of the model as we are providing with contradicting features. On the other hand, the TFIDF handle it with a tricky

mathematical operation, taking the same case where we have common words for both documents. The term frequency is as usual now for inverse document frequency the result of division of number of documents and number of documents containing the particular term will be closer to 1 as the number of documents with term will be closer the total number of the documents, and log of the number closer to 1 will be near to 0. Similarly, for the word which are present in fewer number of documents will have higher value of TFIDF because the value of log of number of documents divided by number of documents with them will be higher. Therefore, using this TFIDF method helps in handling these scenarios very effectively.

Count Vectorizer – Let us see the second vectorizer we have used here. The second vectorizer is a simple but very effective technique called count vectorizer. In this technique the words are converted to vectors based on the frequency of the words i.e., the number of times a word is present in a particular documents or text [32]. This seems a simple method, but it is one of the most common methods because in various application it has shown that the frequency of particular words does contains some useful information about various aspects of the text like sentiment polarity etc.

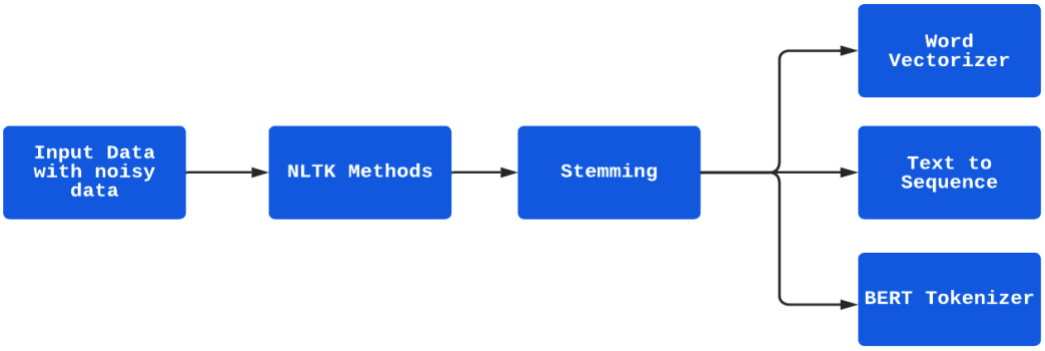


Figure 23: Dataflow Diagram for Data Preprocessing.

4.3.3 The Weak Supervision

In this work, one of the most important parts is to weakly label the data, as we have a large amount of unlabeled data, so in this step we need to generate weak labels and apply those to the entire dataset.

4.3.3.1 Generating Weak Labels

To generate the weak labels, we have taken a different approach; since this is an important process in our cyberbullying detection use case. For the common domains like sentiment analysis, we know the words or the way we express our negative sentiment, so it is easy to find the criteria to generate the weak labels. However, in cyberbullying detection case, the criteria to generate weak labels is complex. So, to find the words or the criteria on the basis with which we can generate the weak labels, we used a small set of fully labeled data and then we first tried to remove all the words that are only present for grammatical reasons and does not contain any information needed to find the class of the text. To find and remove such words we have used polarity as a benchmark .The thought behind it is that if a word has some information, then it will have even little polarity value and it can be negative and positive. So, we have removed all the words that have the zero polarity. Then we plotted the words and found the top few words and selected 8 words which we would use to generate the weak labels.

4.3.3.2 Applying Weak Labels

As a next step in the process, we had to apply the weak labels. For this we have used Snorkel library [40], it has some useful methods that can be used to easily apply the labels to the entire dataset. Also, we have used its labeling function to first generate weak labels based on the 8 words we have selected, by using the basic idea that when these words

are present then the label is 1 (or cyberbullying) else label it as 0, which is not cyberbullying. Also, we have taken one fact into consideration that not all the negative sentiment is cyberbullying but most of the cyberbullying cases have negative sentiments in it. So, we have created one label using polarity of the text, in which we have labeled the text as cyberbullying, when the polarity is negative and labeled the text as not cyberbullying if otherwise. After this process we had total 6 weak labels.

4.3.3.3 Calculating Final Weak Labels

As, we have collected multiple labels, the next step in the process is to compute a final label from all these labels. There are two important ways to calculate the final label, first is the "most vote" method and second is "averaged" method. The description of "most vote" and "averaged" method approaches are explained as follows:

"Most Vote" – In the "most vote" approach, first we calculate the total count of each of the label for each row. In our case, we have two labels: cyberbullying represented by 1 and non-cyberbullying represented by 0. So, taking an example for row 1 ($R_1 1$), we calculated the number of labels with value 1 say N_1^1 and labels with value 0 say N_1^2, then we will see if N_1 is greater than or equal to N_2 i.e. number of cyberbullying labels for R_1 then we will select 1 as the final label for R_1 or else we will take 0 as final label for R_1. We performed similar calculation for each row to find the final label for each row.

N_i^1 = Count of Labels with Label value 1 for i^{th} row

N_i^2 = Count of Labels with Label value 0 for i^{th} row

L_i = Final label for i^{th} row

$$L_i = \begin{cases} 1, & \text{if } N_i^1 \geq N_i^2 \\ 0, & \text{otherwise} \end{cases}$$

Figure 24: Calculation of Final Label Using Most Vote Method.

"Averaged" Method – In the "averaged" method approach, we calculate the average for values of the labels for each row and then we select a threshold value and if the value is more than the threshold value then we take one class as final label or else we take another class as final label. Instead of calculating the "averaged value" directly, by taking into account the concept that each weak label generated from words are having some contradicting information and not too much reliable, and also the concept that the label generated from the polarity is more reliable because for the text which are from the cyberbullying class will have some negative polarity, we use it in our final label calculation. On the basis of it, we have given 40% weightage to the label from the polarity-based label and remaining 60% weightage was divided between 5 labels i.e. 12% each to the remaining five labels. We have also verified it with small amount of fully labeled data with this we have also selected the threshold value of 0.5 for our case.

L_i^1 to L_i^5 = Labels generated from the words for i^{th} row

L_i^6 = Label generated from the polarity for i^{th} row

$$\text{Average} = (L_i^1 + L_i^2 + L_i^3 + L_i^4 + L_i^5) * 0.12 + L_i^6 * 0.4$$

$$L_i = \begin{cases} 1, & \text{if Average} \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

Figure 25: Calculation of Final Label Using Averaged Method.

After calculating the final labels, we have two final labels for each row i.e., one label from “most vote” method and one from the “averaged method”. Then will train each model once with “most vote” label and once with “averaged method” label and evaluate each model compared to the best model. The machine learning algorithms used for our experimentation are explained as follows;

4.3.4 Algorithm Used

We used multiple classification algorithms such as support vector machines (SVM), RNN-LSTM, BERT and distilBERT for detecting cyberbullying. These classification algorithms are explained as follows:

4.3.4.1 Support Vector Machine

A Support Vector Machine is a supervised machine learning algorithm and a discriminating classifier which works by plotting data in a N-Dimensional space and then find the best suitable hyper-plane, which can classify data distinctly into their respective

classes. For finding the hyper-plane, it uses the concept of maximum margin hyper-plane. It can be used for both classification and regression, but it's mostly used for classification problems. SVM uses much lesser computational power than neural networks and it gives very trusted results with both linear and non-linear data.

The main idea on which SVM works is that it tries to find the classifier or decision boundary, such that the distance between decision boundary to the nearest data points of each classes is maximum (such hyper-planes are also known as maximum margin hyper-plane). That's why it's also known as maximum margin classifier.

Margin - A margin can be defined as the distance of the closest points to the decision surface. We can also say that the margin is the distance between the decision boundary and each of the classes. So, in Figure 26 below we can see that points (X_1^1,X_2^1) and (X_1^2,X_2^2) are closest to the decision surface. Hence the distance between these point and decision surface is the Margin. Let's plot above points in the graph below to visualize the concept in more details:

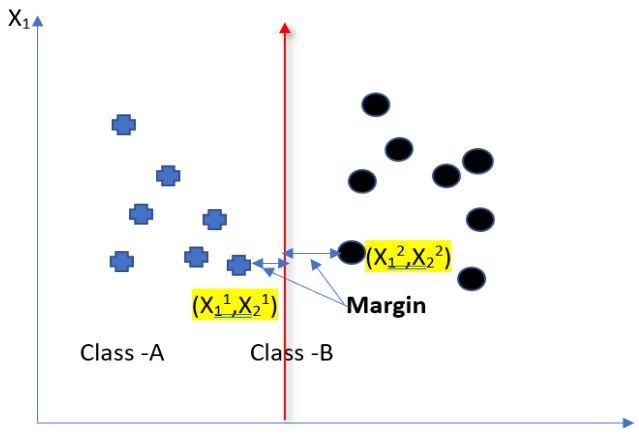


Figure 26: Concept of Margin.

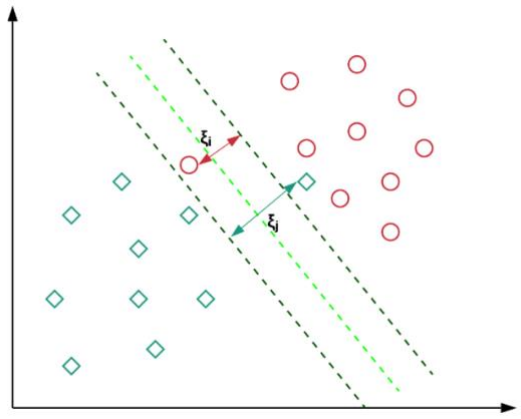


Figure 27: Concept of Hyperplane Separating the Data Points from Both the Classes.

The Figure 27 visualizes the concept of the margin and how margin is calculated using the closest points. The points which are near to the hyper plane are known as support vectors and these points are used to maximize the margin. Any change such as deletion of these support vectors effects the orientation of the hyperplane. To classify our data into two classes: cyberbullying and non-cyberbullying, our first task is to plot the word vectors from both classes in a N-dimensional space, so that SVM can find a hyperplane, which will maximize the margin between both the classes. Following equation 11 shows the cost function for the SVM classifier.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(11)$$

4.3.4.2 RNNLSTM

Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network are very effective algorithm specifically in the field of Natural Language Processing [41]. The RNN has an internal memory. It also has one fundamental aspect that it learns from current and previous computation and is very useful in the sequential data

such as text. Now RNNLSTM is a modified version of the RNN network, and it makes the memory part even simpler. It has three important gates: input gate, output gate and forget gate. The forget gate is very important and it performs the evaluation using current input and the previous hidden state. If the output is 1 then the input is retained and if the output value is 0, it means forget the input value. This is one the most important part of the LSTM architecture.

4.3.4.3 BERT (Bidirectional Encoder Representations from Transformers)

The BERT is a transformer-based machine learning techniques for Natural Language Processing, which was pre-trained and developed by Google. BERT model is based on attention-based mechanism. This mechanism helps the model to select the relevant context of a given word. It encodes the data in very useful manner, and it reads the text from both the directions and thus allows algorithm to have a better understanding of the text. To predict the next word, , it first randomly masks the words in a sentence and then it tries to predict them. To predict the next word, it reads the text from left to right and right to left i.e. it uses full context of the word to predict them.

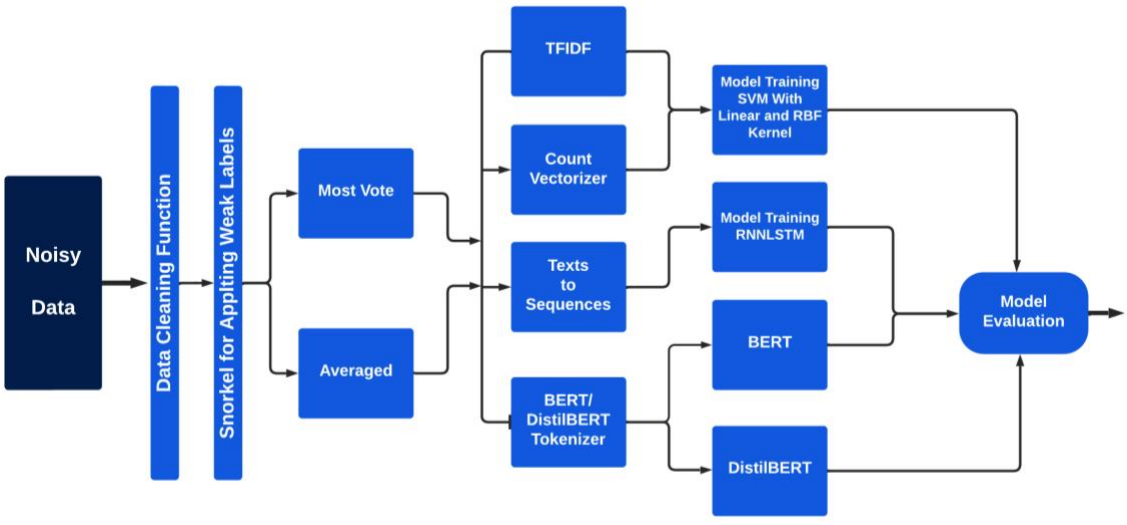


Figure 28: Methodology workflow.

4.3.5 Model Training

After all the data cleaning, data preprocessing, stemming and vectorizer techniques, we have the required data for model training. Further after generating the labels from the "most vote" and "averaged" method we obtain two labels. After that we train SVM, RNNLSTM, BERT and DisltilBERT model with each of these labels.

4.3.5.1 SVM Model Training

We trained the Support Vector Machines (SVM) model with linear kernel and RBF kernel, and we trained SVM with linear kernel with first "most vote" label and then with "averaged" label. Similarly, we trained the SVM model with the linear kernel with "most vote" label and then with "averaged" label. Now with that we have also performed one more experiment where we have trained linear kernel with the output vectors from the TFIDF and output vectors from the count vectorizer. Similarly, we have trained RBF kernel with output from the TFIDF vectorizer and output from count vectorizer. So, for "most vote" label we have trained 4 SVM models and in same way we have trained 4 models for "averaged" label. Then we have performed the model evaluation from the overall process.

4.3.5.1.1 Performance Measurement with Most Vote Label Approach

After training process, we evaluated the model in terms of accuracy, recall, precision, F1 score, bias and "mean square error". We observed that the SVM model did not perform well with the "most vote" label" for both linear kernel and RBF kernel options. SVM model with TFIDF vectors and most vote label gave average results. Accuracy of both the models was around 50%. With respect to other parameters, Recall for RBF kernel was better than the linear kernel and was at 100% and for linear it was 93%. Furthermore, when we consider the Bias and mean square error the linear model has slightly less bias of

42.64% than the RBF kernel with bias of 50% but both the approaches have average mean square error of around 50%.

As shown in the Table 11, in regard to the count vectorizer case with “most vote” label, both the models performed in similar way, with around 50% accuracy, 100% recall, around 50% precision and 50% for bias and 50% for mean square error parameters. Overall, we had an average result with most vote cases for all the kernel and word vectorizer techniques.

Algorithm and Labeling Method	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
Results SVM TFIDF Stem Linear Kernel Most Vote Label	51%	93%	50.5435%	65.493%	42.64%	49%
Results SVM TFIDFStem RBF Kernel Most Vote Label	50%	100%	50%	66.6667%	50%	50%
Results SVM Count Vectorizer Stem Linear Kernel Most Vote	50.5%	100%	50.25%	66.8896%	49.5%	49.5%
Results SVM Count Vectorizer Stem RBF Kernel Most Vote	50%	100%	50%	66.6667%	50%	50%

Table 11: Comparison of Results of SVM Model with Most Vote Label.

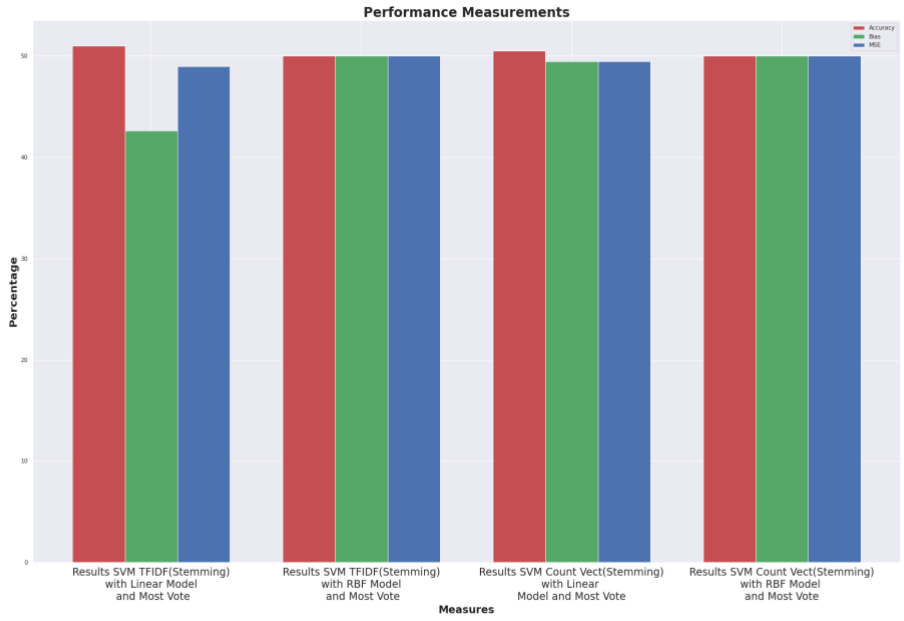


Figure 29: Comparison of Results of all the SVM model with Most Vote Label.

4.3.5.1.2 Performance Measurement with Averaged Label Approach

Next, we have trained all the SVM models with TFIDF and count vectorizer vectors using averaged label option. Similar to the previous process we evaluated the model in terms of accuracy, recall, precision, F1 score, bias and mean square error. In this scenario too, the SVM models performed in a similar manner as shown in the Table 12. After analyzing the results, we found that, the accuracy of both the models was around in range of 50% to 51.5%. So, we can say that these are not promising results. However, with other parameters, Recall for RBF kernel was better than the linear kernel and was at 99%. For the precision metric, all the SVM models trained with averaged label for different kernels and word vectorizer techniques, the results were in the range of 50% to 50.8%. For recall metric, all the models performed in same manner and gave results in the range of 88% to 92%. For bias and mean square error metric, all the models performed almost equally and gave a value of around 50%. For the model trained with linear kernel, TFIDF word vectorizer and averaged label performed slightly better than other models and gave result of 38%. However, SVM model with RBF kernel and TFID vectorizer output gave much higher bias of 49%.

Based on these results, we observed that SVM model training with averaged level approach was not too much promising.

Algorithm and Labeling Method	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
Results SVM TFIDF Stem Linear Kernel Averaged Label	51.5%	88%	50.8671%	64.4689%	0.3832%	48.5%
Results SVM TFIDF Stem RBF Kernel Averaged Label	50%	99%	50%	66.443%	49.01%	50%
Results SVM Count Vectorizer Stem Linear Kernel Averaged Label	50%	92%	50%	64.7887%	42.64%	50%
Results SVM Count Vectorizer Stem RBF Kernel Averaged Label	50.5%	92%	50.2732%	65.0177%	42.22%	49.5%

Table12: Comparison of Results of SVM Model with Averaged Label.

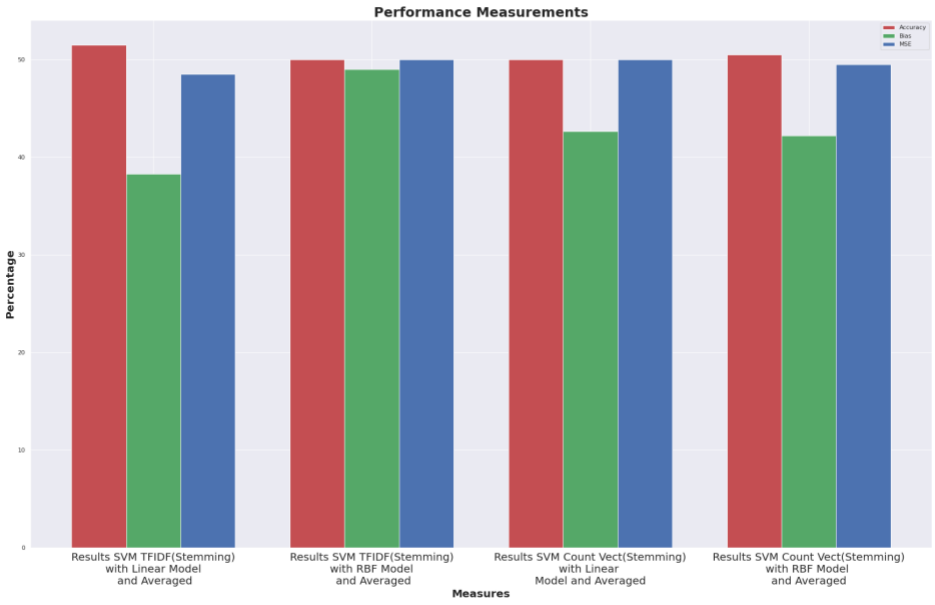


Figure 30: Comparison of Results of all the SVM Models with Averaged Label.

4.3.5.2 RNNLSTM Model Training

After experimenting with SVM model, we switched our focus to more advanced models like RNNLSTM. We trained the RNNLSTM models with first "most vote" label and then with "averaged" label. We will analyze the RNNLSTM model training results as follows:

4.3.5.2.1 RNNLSTM Model Training with "Most Vote" Label

We trained the RNNLSTM model with "most vote" approach first. Training the model with "most vote" approach, resulted in promising results, with an accuracy of the 97.44%, recall of 99.31%, precision of 97.89% results. Bias of model and mean square error of model also reduced to 8.51% and 2.56% respectively. Overall, the RNNLSTM model performed very well on all the parameters.

4.3.5.2.2 RNNLSTM Model Training with "Averaged" Label

After getting some reliable results from the RNNLSTM and "most vote" label, we trained the RNNLSTM model with "averaged" label approach, then we analyzed the results and compared this with most vote model to see which model worked more accurately. We found that the RNNLSTM model with averaged label worked very well and gave some promising results. When compared to the model trained with "most vote" label approach, the "averaged label" approach shows slightly less results. It resulted in an accuracy of 95.97%, recall of 97.31%, and precision of 96.33%. Also, a bias of 22.63% is slightly higher than the most vote model. Although it gave decent mean square error of 4.03%, it is slightly higher than model trained with most vote label.

4.3.5.3 BERT Model Training

After getting some good results from RNNLSTM model for both most vote and averaged label, for BERT model we have used the BERT tokenizer with the cleaned data and not the stemmed data, we evaluated the model on different parameters like accuracy, bias, mean square error, Recall and precision for most vote and averaged label.

4.3.5.3.1 BERT Model Training with "Most Vote" Label Approach

We first trained the BERT model with the "most vote" label approach. Here the model gave some decent results with 78.5% accuracy, 76.11% recall, 100% precision, and 86.4353% F1 score. The model also resulted in bias of the 13.62% and mean square error of 21.5%. These results indicate BERT approach performed poorly when compared to RNNLSTM models.

4.3.5.3.2 BERT Model Training with "Averaged" Label Approach

As a next step we trained the BERT model with averaged label approach, and it gave much better results compared to BERT model trained with most vote label. This approach gave 90.5% accuracy, 86.1538% recall, 99.115% precision, and 92.1811% F1 score. The model also resulted bias of the 23.47% and mean square error of 9.5%. Here in terms of bias this model is little in lower side while the bias of the model is better compared to "most vote" label approach. This approach also performed poorly when compared to RNNLSTM models.

4.3.5.4 DistilBERT Model Training

In the next step of experimentation we then trained DistilBERT model to evaluate its performance. For DistilBERT model, we used the DistilBERT tokenizer with the cleaned data again and not the stemmed data. Then we evaluated the model on different parameters such as accuracy, bias, mean square error, Recall and precision for "most vote" and averaged label.

4.3.5.4.1 DistilBERT Model Training with "Most Vote" Approach

We first trained the DistilBERT model with the "most vote" label approach, we observed some downfall in the performance compared to RNNLSTM approach. The model gave 73% accuracy, 70% recall, 100% precision, and 82.3529% F1 score. Bias and means square error, were observed at 16.29% and 27% respectively. The results are inferior compared to RNNLSTM, as well both the BERT models.

4.3.5.4.2 DistilBERT Model Training with "Averaged" Label Approach

In this experiment we trained the DistilBERT model with the averaged label approach. Here model performed with very good results and resulted in accuracy of 99%, 98.46% recall, 100% precision, and 99.2248% F1 score. It resulted in higher bias of

22.76% but very lower mean square error. The model seems to be working as good as RNNLSTM model but the only issue is higher bias.

Algorithm and Labeling Method	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
Results RNNLSTM with Most Vote	97.4436%	99.3178%	97.8921%	98.5998%	8.51%	2.56%
Results RNNLSTM with Averaged Label	95.9733%	97.3173%	96.5526%	96.9335%	22.63%	4.03%
Results BERT with Most Vote	78.5%	76.1111%	100%	86.4353%	13.62%	21.5%
Results BERT with Averaged Label	90.5%	86.1538%	99.115%	92.1811%	23.47%	9.5%
Results DistiltBERT with Most Vote	73%	70%	100%	82.3529%	16.29%	27%
Results DistiltBERT with Averaged Label	99%	98.4615%	100%	99.2248%	22.76%	1%

Table 13: Comparison of Results of RNNLSTM, BERT and distilBERT Models with Most Vote and Averaged Label.

4.4 Model Evaluations and Discussion of Results

After training all the models and evaluating the results, we can conclude that the SVM models with any label did not perform well and cannot be considered as a reliable technique. Also, in this work we were more focused on the fairness of the model, and all the SVM model approaches did not perform well and are highly biased. Thus, we can conclude that these SVM-based models are not fair. However, RNNLSTM models performed well with both “most vote” and “averaged” label, and the bias seems to as minimum as 8.51% with the mean square error of 2.56% with RNNLSTM for most vote label approach. Also, the RNNLSTM model with averaged label approach works equally well but it has higher bias. BERT models performed decently but it did not perform as well as the RNNLSTM models. DistilBERT model with most vote did not perform well but performed better compared to SVM models. DistilBERT model with “averaged” label performed better than any other model but it had higher bias compared to RNNLSTM model trained with “most vote” label. So, after overall comparison we conclude that the

RNNLSTM model trained with “most vote” model performed better than all other models, when we consider all the other parameters as shown in Table 14, Figures 31 and 32.

Algorithm and Labeling Method	Accuracy	Recall	Precision	F1 Score	Bias	Mean Squared Error
Results SVM TFIDF Stem Linear Kernel Most Vote Label	51%	93%	50.5435%	65.493%	42.64%	49%
Results SVM TFIDFStem RBF Kernel Most Vote Label	50%	100%	50%	66.6667%	50%	50%
Results SVM Count Vectorizer Stem Linear Kernel Most Vote	50.5%	100%	50.25%	66.8896%	49.5%	49.5%
Results SVM Count Vectorizer Stem RBF Kernel Most Vote	50%	100%	50%	66.6667%	50%	50%
Results SVM TFIDF Stemming Linear Kernel Averaged	51.5%	88%	50.8671%	64.4689%	38.32%	48.5%
Results SVM TFIDF Stemming RBF Kernel Averaged	50%	99%	50%	66.443%	49.01%	50%
Results SVM Count Vectorizer Stemming Linear Kernel Averaged	50%	92%	50%	64.7887%	42.64%	50%
Results SVM Count Vectorizer Stemming RBF Kernel Averaged	50.5%	92%	50.2732%	65.0177%	42.22%	49.5%
Results RNNLSTM with Averaged Label	95.9733%	97.3173%	96.5526%	96.9335%	22.63%	4.03%
Results RNNLSTM Most Vote	97.4436%	99.3178%	97.8921%	98.5998%	8.51%	2.56%
Results DistiltBERT with Averaged Label	99%	98.4615%	100%	99.2248%	22.76%	1%
Results DistiltBERT Most Vote	73%	70%	100%	82.3529%	16.29%	27%
Results BERT Most Vote Label	78.5%	76.1111%	100%	86.4353%	13.62%	21.5%
Results BERT Averaged Label	90.5%	86.1538%	99.115%	92.1811%	23.47%	9.5%

Table 14: Comparison of SVM, RNNLSTM, BERT and DistilBERT Models with “Most Vote” and “Averaged” Label.

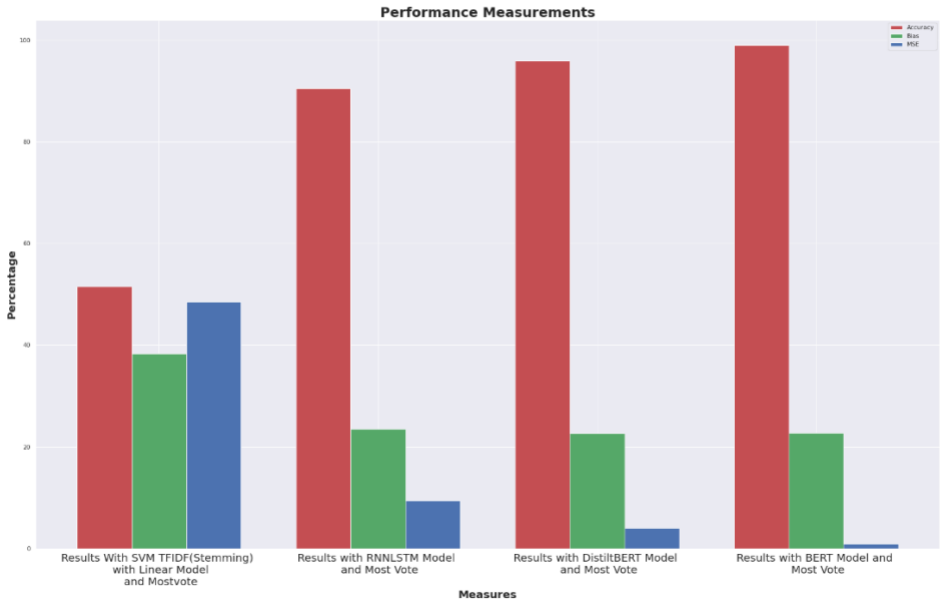


Figure 31: Comparison of SVM, RNNLSTM, BERT and DistilBERT Models with “Most Vote” Label.

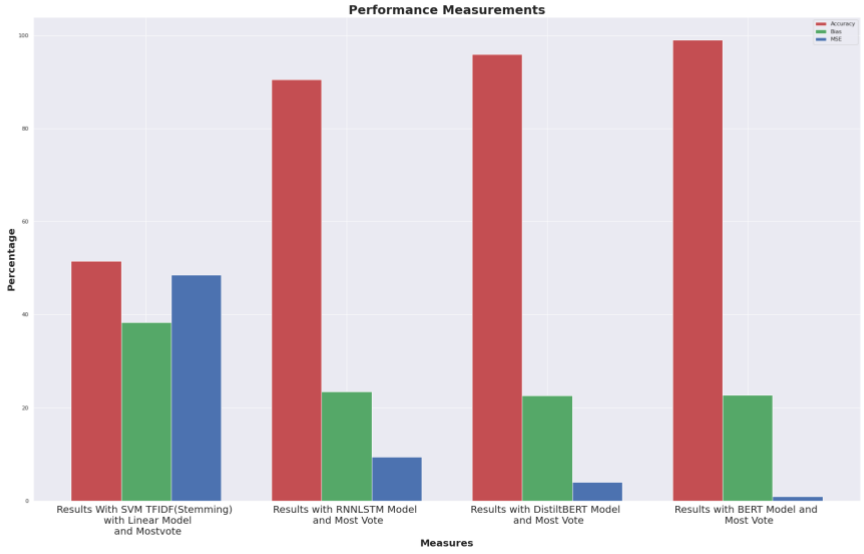


Figure 32: Comparison of SVM, RNNLSTM, BERT and DistilBERT Models with Averaged Label.

4.5 Comparison of Results of Weakly Supervised Learning and Fully Supervised Learning

In this section, we evaluate the performance of fully supervised learning approach compared to weakly supervised learning approach using different metrics explained in earlier section. This comparison is very crucial because if we can achieve similar results with weakly supervised learning, then it will help researchers and practitioners to apply weakly supervised machine learning to different domains, where there are limitations of availability of fully labeled data.

For fully supervised learning technique, we used the fully labeled data, where the dataset contains both feature set and target values. Further, we have used same models: SVM, RNNLSTM, BERT and distilBERT for both weakly supervised learning and fully supervised learning. Only difference between fully supervised learning technique and the

weakly supervised learning technique is in the target label. For fully supervised learning, we have clean labels already present in the dataset and for weakly supervised learning we calculate weak labels using our approach described in section 4.3.3.

Underlying Algorithm Used	Accuracy with Fully Supervised Learning	Recall with Fully Supervised Learning	Precision with Fully Supervised Learning	F1 Score with Fully Supervised Learning	Bias with Fully Supervised Learning	Mean Squared Error with Fully Supervised Learning	Accuracy with Weakly Supervised Learning	Recall with Weakly Supervised Learning	Precision with Fully Supervised Learning	F1 Score with Weakly Supervised Learning	Bias with Weakly Supervised Learning	Mean Squared Error with Weakly Supervised Learning
Results with SVM and TFIDF with stem linear	80.4341%	84.8729%	77.9613%	81.2704%	25.2%	19.57%	51%	93%	50.5435%	65.493%	42.64%	49%
Results with RNNLSTM (Most Vote label for Weakly Supervised Learning)	78.6412%	79.4752%	77.9412%	78.7007%	25.01%	21.36%	97.4436%	99.3178%	97.8921%	98.5998%	85.1%	2.56%
Results with DistiltBERT (Averaged label for Weakly Supervised Learning)	90%	88.8889%	90.7216%	89.7959%	25.01%	10%	99%	98.4615%	100%	99.2248%	22.76%	1%
Results with BERT (Averaged label for Weakly Supervised Learning)	89.4%	91.7355%	87.0588%	89.336%	25.04%	10.6%	90.5%	86.1538%	99.115%	92.1811%	23.47%	9%

Table 15: Comparison of Results for Weakly Supervised Learning and Fully Supervised Learning.

Table 15 shows the comparison of results of various parameters for fair and accurate cyberbullying detection. As shown in the Table 15, the weakly supervised learning has performed better than fully supervised learning, for all classification algorithms such as RNNLSTM, BERT and DistilBERT except SVM models. For SVM models the fully supervised learning works better. For example, the mean square error and bias for the weakly supervised learning for RNNLSTM, DistilBERT and BERT is low and also has a very low bias of 2.56%, 1% and 9.5% respectively. This shows that model is fair and on top of that the weakly supervised learning even works better if consider other detection parameters like accuracy, precision, recall and F1 score. This demonstrate that, the weakly supervised learning works efficiently and even better than fully supervised learning for detecting cyberbullying. So, it further establishes our study that if we use more advanced and highly pretrained model then even with weakly supervised learning we can develop a very reliable, fair and efficient model. Thus, it gives us ability to extend the uses of machine learning in different domain even with limited or no fully labeled data.

4.6 Research Contribution

In this work we have developed a very efficient model to detect cyberbullying using weak supervision. Further, we have also compared the results obtained using weak supervision with the results obtained using fully supervised learning in chapter III to find if we have achieved similar results using weak supervision, after comparison we found that we obtained similar results using weak supervision in terms all the performance measurement parameters such as accuracy, precision, recall, F1 score, bias and mean square error. This establishes the fact that we can build efficient algorithm to detect cyberbullying using weak supervision itself. Further we have used a different method to calculate the final label using "averaged" method using small amount of fully labeled data, the method can be used in different domains where we have small amount of cleaned label to improve the efficiency of the model. Overall, it provides an efficient model to detect cyberbullying using weak supervision.

4.7 Future Work Recommendation

In our experimentations RNNLSTM and DistilBERT model results are much reliable and encouraging. As our current work only focused on only changing the underlying algorithm for evaluation purposes. Further work can be done to remove model bias and anomalies from the data. As a result, combining the techniques to reduce model bias and reducing bias from data side, we can even achieve some better results. Also, the proposed approach can be utilized in other domains such as detection of fake news and hate speech detection.

4.8 Conclusion

Cyberbullying is one of the pressing issues in our society and it is growing at very high rate due to rapid increase in social media users. So, in this work we have presented a very fair and reliable model to detect cyberbullying from social media data. In this work, we have also established the fact that if we use more advanced algorithm like RNNLSTM, BERT and DistilBERT we can build a very effective algorithm even with the weak supervision concept. Through experiments, we have also established the fact that if we can do some analysis while calculating the final label using “most vote” and “averaged” labeling technique, then we can further improve the cyberbullying detection results. The results here also provide a very strong ground for its use in different domains and it also lays ground for future work to remove further noise and bias in the model. This will help us to build even more trustable system that can be used to control and handle many issues of our society due to digitalization such as cyberbullying.

CHAPTER V

LESSONS LEARNT, FUTURE WORK RECOMMENDATIONS AND CONCLUSION

5.1 Lessons Learned:–

There are several lessons learnt during the thesis work with respect to machine learning and cyberbullying detection. These are as follows:

(a) Firstly, we learnt that cyberbullying is a very critical issue of our society, and there are many complexities in detecting it from the social media data, because it is similar to the case of sentiment analysis and thus, we need to have more advanced techniques and machine learning algorithms that can effectively differentiate between negative sentiment and cyberbullying.

(b) Another learning which we gained is that the traditional support vectors machines are effective in fully supervised learning, but it fails drastically in case of weakly supervised learning. On the other hand, the advanced model like RNNLSTM, BERT and distilBERT are very effective in both fully supervised learning and weakly supervised learning.

(c) Further we have also learnt that RNNLSTM is a very effective algorithm in both fully and weakly supervised learning. RNNLSTM worked even better than the highly pre-trained BERT and distilBERT models. This is because we have trained RNNLSTM with entirely our dataset and due to its memory units, which stores the previous inputs and this feature helps the model to understand the linguistic feature of the text and the linguistic characteristic of any text is a powerful feature for extracting useful information such as sentiment or cyberbullying.

(d) During the process of generating weak labels and calculation of final labels, we learnt that if we can perform some analytics on the multiple weak labels and using that if we assign weights to each label and then calculate the average, then it performs much better than the simple averaging method, which gives equal weightage to each label.

(e) It is important to evaluate any classification machine learning algorithm not only using 'accuracy' parameter but also based on multiple parameters such as bias , mean square error, recall, precision parameters. These parameters are an important measure of bias and fairness to help us in evaluating and selecting the best model.

(f) One important learning from this work is that with weak supervision we can build an effective scalable model. Compared to fully supervised learning techniques, using weakly supervised learning techniques, in this work, we have demonstrated that with few processes in data preprocessing and using state of the art machine learning models, we can develop an effective algorithm which is not only high in accuracy but also has low bias and high fairness.

5.2 Future Work Recommendations

The core idea of weakly supervised learning and its successful use in the detection of cyberbullying opens the scope of implementation in many other NLP domains. We see many issues related to cyberbullying, which has come up due to the spread of digitalization and most of these cases can be detected using natural language processing techniques. Therefore, we can use the techniques presented in this work to detect these issues and remove it from social media platforms. Further, we have also presented a simpler mathematical equation to calculate final label using the averaging technique and this equation can be used to implement the weakly supervised learning techniques for the other use cases in other domains . This work can also be extended to remove bias from data to further improve the model in terms of bias and fairness. The successful implementation of weak supervision n this work further opens doors for research suing weak supervision in many other critical domains where we do not have enough fully labeled data. Overall, this work has presented some useful ideas that can be leveraged in other machine learning and natural language processing use cases.

5.3 Conclusion

Cyberbullying is one of the most critical social issues which is spreading rapidly in recent times due to the emergence of social media. Therefore, in this work we have focused on detecting the cyberbullying using two types of machine learning techniques: weakly supervised learning and fully supervised learning.

In chapter 2, we described the experiments using weakly supervised learning techniques and their results. The main reason behind conducting these experiments is that we wanted to establish the fact that even though detection of cyberbullying is a complex

task, using weakly supervised learning and advanced algorithms, we can perform the cyberbullying detection with not only a decent accuracy, bias and fairness, but also it removes the hurdle of labeling large scale of data to build such methods.

In chapter 3, we escribed the similar task of cyberbullying detection using fully supervised learning techniques, that can serve as a benchmark for weakly supervised learning. We evaluated the performance of the supervised learning techniques in cyberbullying using parameters such as accuracy, recall, precision, F1score, mean square error and bias to verify performance of the models in terms of bias, fairness and classification accuracy.

Finally, in chapter 4, we described the experiments with weakly supervised learning with some changes in process of calculation of final label using averaged technique. In these experiments, we trained the models with new state-of-the-art algorithms and compared the results of all experiments of weakly supervised learning with fully supervised learning. These performance evaluation experiments helped us to determine if we can achieve similar results for cyberbullying detection with weakly supervised learning techniques when compared to fully supervised learning techniques.

The comparison of results from both fully supervised learning and weakly supervised learning techniques established the fact that using weakly supervised learning, we can achieve equivalent results which we have got from fully supervised learning. Thus, we can build effective systems which are efficient in terms of accuracy, bias and fairness in detecting cyberbullying without spending efforts and cost on labeling of data. This lays a strong foundation for researchers and practitioners to apply this methodology in different languages to detect cyberbullying as well as other use cases in different domains with

unlabeled data by using a powerful combination of weakly supervised learning and latest advanced algorithms.

REFERENCES

[1] M. Chu "The Origin of Cyberbullying + 5 Ways to Identify and Prevent It," *dataoverhaulers.co*.[Online] Available: dataoverhaulers.com https://dataoverhaulers.com/origin-of-cyberbullying/ [Accessed: Sep. 21, 2021].

[2] UNICEF poll, "'More than a third of young people in 30 countries report being a victim of online bullying" *unicef.org*, [Online] Available: https://www.unicef.org/press-releases/unicef-poll-more-third-young-people-30-countries-report-being-victim-online-bullying [Accessed: Sep. 21, 2021].

[3] "Online Safety & Parents" *services.google.com*.[online]. Available: https://services.google.com/fh/files/blogs/parent_teacher_survey_us.pdf [Accessed: Sep. 21, 2021].

[4] Kai Shu, Guoqing Zheng , Yichuan Li , Subhabrata Mukherjee , Ahmed Hassa, Awadallah, Scott Ruston and Huan Liu "Leveraging Multi-Source Weak Social Supervision for Early Detection of Fake News" in European Conference on Machine Learning and Knowledge Discovery in Databases., Frank Hutter, Kristian Kersting, Jefrey Lijffijt, Isabel Valera, Eds. 2021, pp. 650-666.

[5] Bandeh Ali Talpur and Declan O'Sullivan "Cyberbullying severity detection: A machine learning approach." *journals.plot.org*, Oct. 2020. [Online] Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0240924 [Accessed: Sep. 21, 2021].

[6] Md Manowarul Islam, Md Ashraf , Linta Arnisha Akter , Selina Sharmin, Uzzal Kumar Acharjee, "Cyberbullying Detection on Social Networks Using Machine Learning Approaches" in Asia-Pacific Conference on Computer Science and Data

Engineering (CSDE), Gold Coast, Australia, 2020, doi: 10.1109/CSDE50874.2020.9411601.

[7] A. Muneer and S. M. Fati, “A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter,” Futur. Internet, vol. 12, no. 11, 2020, doi: 10.3390/fi12110187.

[8] Dadvar, Maral & Eckert, Kai. (2018). Cyberbullying Detection in Social Networks Using Deep Learning Based Models; A Reproducibility Study. 10.13140/RG.2.2.16187.87846.

[9] Elaheh Raisi and Bert Huang “Cyberbullying Detection with Weakly Supervised Machine Learning,” in Proc. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining., July 2017, pp. 409–416.

[10] Jason Wang, Kaiqun Fu, Chang-Tien Lu, November 12, 2020, "Fine-Grained Balanced Cyberbullying Dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/kn1c-zx22.

[11] J. Wang, K. Fu and C. -T. Lu, "SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection," IEEE International Conference on Big Data (Big Data), 2020, pp. 1699-1708, doi: 10.1109/BigData50022.2020.9378065.

[12] S. Agrawal and A. Awekar, “Deep learning for detecting cyberbullying across multiple social media platforms,” in *European Conference on Information Retrieval*. Springer, 2018, pp. 141–153.

[13] U. Bretschneider, T. Wohner, and R. Peters, “Detecting online harassment in social networks,” in *ICIS*, 2014.

[14] D. Chatzakou, I. Leontiadis, J. Blackburn, E. D. Cristofaro, G. Stringhini, A. Vakali, and N. Kourtellis, "Detecting cyberbullying and cyberaggression in social media," *ACM Transactions on the Web (TWEB)*, vol. 13, no. 3, pp. 1–51, 2019.

[15] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," *arXiv preprint arXiv:1703.04009*, 2017.

[16] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the NAACL student research workshop*, 2016, pp. 88–93.

[17] Z. Waseem, "Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter," in *Proceedings of the first workshop on NLP and computational social science*, 2016, pp. 138–142.

[18] J.-M. Xu, K.-S. Jun, X. Zhu, and A. Bellmore, "Learning from bullying traces in social media," in Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies, 2012, pp. 656–666.

[19] Hinduja, S. & Patchin, J. W. (2021). "Cyberbullying Identification, Prevention, and Response. Cyberbullying Research Center (cyberbullying.org)". Available: https://cyberbullying.org/Cyberbullying-Identification-Prevention-Response-2021.pdf

[20] Khurana, D.; Koli, A.; Khatter, K.; Singh, S. Natural language processing: State of the art, current trends and challenges. arXiv2017, arXiv:1708.05148.

[21] IBM Cloud Education, "Natural Language Processing (NLP)" *IBM Cloud*

Education. [Online]. Available: https://www.ibm.com/cloud/learn/natural-language-processing [Accessed: Sep. 21, 2021].

[22] Pierre Lison, Jeremy Barnes, and Aliaksandr Hubin. "skweak: Weak Supervision Made Easy for NLP," arXiv.org, 2104.09683, 2021. [Online]. Available: https://arxiv.org/abs/2104.09683 [Accessed: Sep. 21, 2021].

[23] "Distilroberta-base,"huggingface.co.[online]. Available: https://huggingface.co/distilroberta-base[Accessed: Sep. 21, 2021].

[24] Sokolova, M., N. Japkowicz, and S. Szpakowicz. 2006. "Beyond Accuracy, F-score and ROC: A Family of Discriminant Measures for Performance Evaluation." In AI 2006: Advances in Artificial Intelligence, edited by A. Sattar and B.-H. Kang

[25] "Fairness (machine learning)," *wikipedia.org* [Online] Available: https://en.wikipedia.org/wiki/Fairness_(machine_learning). [Accessed Mar. 20, 2022].

[26] Oneto, Luca and Chiappa, Silvia "Fairness in Machine Learning," *arXiv.org,* 2012.15816 cs.LG. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2020arXiv201215816O/abstract [Accessed: Mar. 20, 2022].

[27] John Chen et al "Exploring Text Specific and Blackbox Fairness Algorithms in Multimodal Clinical NLP," *arXiv.org,* 2011.09625 cs.CL. [Online]. Available: https://arxiv.org/abs/2011.09625. [Accessed: March. 20, 2022].

[28] Ninareh Mehrabi "A Survey on Bias and Fairness in Machine Learning" *arXiv.org,* 1908.09635 cs.LG. [Online]. Available: https://arxiv.org/abs/1908.09635. [Accessed: March. 20, 2022].

[29] S. Bird and E. Loper. “NLTK: The natural language toolkit. In Proc. of the 1st ACL Interactive Poster and Demonstration Sessions”, Barcelona: ACL, Jul. 2004, pp 214–217.

[30] S. Qaiser and R. Ali, “Text mining: use of tf-idf to examine the relevance of words to documents,” *International Journal of Computer Applications,* vol. 181, no. 1, pp. 25–29, 2018.

[31] Vadim Andreevich Kozhevnikov and Evgeniya Sergeevna Pankratova, “Research Of The Text Data Vectorization and Classification Algorithms Of Machine Learning”, ISJ Theoretical & Applied Science, 05 (85), 574-585

[32] Choucalas, Vida Zoe, “Cyberbullying and How It Impacts Schools,” Indiana State University. [Online]. Available: https://scholars.indstate.edu/bitstream/handle/10484/8056/Barker-Choucalas,%20Vida.pdf?sequence=2 [Accessed: Mar. 20, 2022].

[33] A. M. Rahmani et al., “Machine Learning (ML) in Medicine: Review, Applications, and Challenges,” Mathematics, vol. 9, no. 22, p. 2970, Nov. 2021, doi: 10.3390/math9222970.

[34] Z.-H. Zhou, “A brief introduction to weakly supervised learning,” National Science Review, vol. 5, no. 1, pp. 44–53, 2017.ß

[35] Grover, A., Choi, K., Shu, R. and Ermon, S., 2022. “Fair Generative Modeling via Weak Supervision.” *arXiv.org*, 1910.12008 cs.LG. [online]. Available: https://arxiv.org/abs/1910.12008v1 [Accessed 17 July 2022].

[36] Kai-Wei Chang, Vinod Prabhakaran, and Vicente Ordonez. 2019. “Bias and fairness in natural language processing”, in Proceedings of the 2019 Conference on

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP):Tutorial Abstracts, Hong Kong, China. Association for Computational Linguistics.

[37] Heckerman, D. 1995a. A Bayesian Approach to Learning Causal Networks. Technical report MSR-TR-95-04, Microsoft Research.

[38] Ratner, A. et al. Snorkel: Rapid training data creation with weak supervision. Proceedings VLDB Endowment 11, 269–282 (2017).

[39] Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network *arxiv.org.* 2022. [online] Available: https://arxiv.org/pdf/1808.03314.pdf [Accessed 17 July 2022]

[40] S. Chaturvedi, V. Mishra and N. Mishra, "Sentiment analysis using machine learning for business intelligence," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017, pp. 2162-2166, doi: 10.1109/ICPCSI.2017.8392100.

[41] Cyberbullying Dataset *Kaggle.com.* [online] Available: https://www.kaggle.com/datasets/saurabhshahane/cyberbullying-dataset [Accessed 17 July 2022].

[42] Chiappa, S. and Isaac, W., 2019. A Causal Bayesian Networks Viewpoint on Fairness. Privacy and Identity Management. Fairness, Accountability, and Transparency in the Age of Big Data, pp.3-20.