MCDefender: Toward Effective Cyberbullying Defense in Mobile Online Social Networks

Nishant Vishwamitra
Clemson University
Clemson, SC 29634
nvishwa@clemson.edu

Xiang Zhang
Clemson University
Clemson, SC 29634
xzhang7@clemson.edu

Jonathan Tong*
Clemson University
Clemson, SC 29634
jxtdragon@gmail.com

Hongxin Hu
Clemson University
Clemson, SC 29634
hongxih@clemson.edu

Feng Luo
Clemson University
Clemson, SC 29634
luofeng@clemson.edu

Robin Kowalski
Clemson University
Clemson, SC 29634
rkowals@clemson.edu

Joseph Mazer
Clemson University
Clemson, SC 29634
jmazer@clemson.edu

ABSTRACT

Cyberbullying in Online Social Networks (OSNs) has emerged as one of the most severe social concerns. Cyberbullying can be described as a form of bullying where a perpetrator uses electronic means to cause harm to a victim. With the proliferation of smartphone technology in present times, there has been a steady shift in the usage of OSNs from traditional computers to mobile devices. However, existing systems that defend against cyberbullying are largely applicable only to traditional computing platforms and cannot be directly applied to detect cyberbullying in mobile platforms. To address such a critical issue, we investigate an innovative mobile cyberbullying defense system called *MCDefender* that can effectively detect and prevent cyberbullying in mobile OSNs. We first analyze the key challenges that differentiate cyberbullying conditions in traditional and mobile platforms. We then investigate a two-level detection mechanism for comprehensive cyberbullying detection in mobile OSNs where cyberbullying can be quickly detected before a cyberbullying message is sent through a mobile device and hidden cyberbullying attacks can be also detected through a more fine-grained and context-aware analysis. To demonstrate the feasibility of our approach, we implement and evaluate an Android application based on *MCDefender*. Our evaluation results show that our mobile application can detect cyberbullying with a high accuracy of 98.9% for OSNs.

Keywords

Cyberbullying defense; Social networks; Deep learning; Pronunciation

1. INTRODUCTION

According to a recent study by the Pew Research Center [18], 92% of teens (Age 13-17) spend time online today. Cyberbullying has emerged as one of the most serious social issues due to the rise of people accessing the Internet [18]. Cyberbullying can be defined as an aggressive, intentional act conducted by either a group or an individual in cyberspace using information and communication technologies (e.g. e-mail, mobile phone, and social network) repeatedly or over time against victims who cannot easily defend themselves [19]. Online Social Networks (OSNs) is one platform where cyberbullying incidents are frequently reported, due to the fact that OSNs provide an uninhibited and unsupervised space for users to interact with each other.

IWSPA' 17, March 24 2017, Scottsdale, AZ, USA.

DOI: http://dx.doi.org/10.1145/3041008.3041013

Sample	Examples	Noise Type
1	"! w@nN@ l!qqhH+ y0 d!(K 0N f!r3 N d3nN sM0k3 *!t w!+ m@ v@q!n@*"	Symbols
2	"wHy yUhH w0N+ fU(K m3 !N d@ @$$ h0l3 ???"	Symbols
3	"lol yew on sum otha shxt nd not even dressed in all black"	Intended typos
4	"im sur3 sh3 d0nt want y0u"	Numbers
5	"iloveyourpenis"	Concatenation

Table 1. Examples of Noisy Data

A new trend has emerged with more users changing the way they access OSNs from traditional computers to mobile devices. In particular, 73% of the adolescents who use the Internet are *mobile internet users* [18], who access the Internet using mobile devices such as smartphones and tablets. The convenience of mobile devices coupled with their ease of use makes accessing OSNs more suitable to mobile devices. Cyberbullying on mobile devices distinctly differs from cyberbullying in traditional computers. In traditional computers, users have a physical keyboard to post messages to social networks. However, the input in mobile devices is through a soft on screen keyboard, which makes the input data *noisy* [21]. Table 1 demonstrates several examples of such noisy data. From Table 1, we can observe that there are several different types of noisy data. The Symbols type (e.g. samples 1 and 2) refers to the use of symbols that look like alphanumeric characters to substitute alphanumeric characters in the messages. Similarly, the Numbers noise type (e.g. sample 4) uses digits to substitute letters in messages. Concatenation type (e.g. sample 5) refers to words that are not separated by spaces, despite of which it is still readable by human beings. They can be classified as *unintended* typos as the objective is not to hide the true meaning of words. Although some algorithms [5, 9, 11] have been designed to defend against cyberbullying attacks in traditional computing platforms, there are no effective algorithms specifically designed for mobile social networks that can be used to deal with noisy data entered through mobile devices.

* Intern from D.W. Daniel High School, Central, SC USA

In this paper, we propose a novel mobile cyberbullying defense system called *MCDefender* for effective detection of cyberbullying in mobile OSNs as well as dynamic intervention in case of cyberbullying incidents. Our system addresses the above challenges by employing a unique *two-level detection* mechanism. For detecting potential cyberbullying before messages are posted, a *Pre-Sending Quick Detection* approach is designed in *MCDefender*. This acts as a deterrent to the perpetrator from committing the act of cyberbullying. We develop a RegEx search engine that performs the quick detection of cyberbullying by performing fuzzy comparisons with a dictionary of cyberbullying regular expression strings, with the capability of performing quick detection of cyberbullying in noisy data. For a more fine-grained and context-aware detection of cyberbullying after posting messages, *MCDefender* uses a *Pronunciation-based Convolutional Neural Network (PCNN)* architecture that uses a pronunciation-based approach for pre-processing cyberbullying data. We observe that our pronunciation based approach works especially well in detecting noisy cyberbullying content in mobile systems. To evaluate our system, we have implemented an Android application for cyberbullying defense in mobile social networks. We have recorded an accuracy of 98.9% for PCNN in our experiments with a Twitter dataset [11] and an accuracy of 92.7% for our RegEx mechanism on that dataset. To the best of our knowledge, *MCDefender* is the *first* mobile cyberbullying defense system that can effectively defend cyberbullying both before and after sending messages to OSNs considering unique features of mobile devices.

The rest of the paper is organized as follows. Section 2 presents the problem statement for our work. Section 3 gives an overview of *MCDefender* system. Section 4 presents our *two-level detection* mechanism and how this mechanism performs well in mobile OSNs. In Section 5, we discuss our dynamic intervention mechanism. Section 6 discusses the implementation and evaluation of our mobile app. In Section 7, we present related work. We conclude this paper along with our future work in Section 8.

2. PROBLEM STATEMENT

Research in the field of cyberbullying detection mostly focus on detecting cyberbullying on a 'per-message' basis [3, 5, 10, 11], which is not as efficient as a cyberbullying detection system that uses the complete context of a post to detect cyberbullying. The concept of cyberbullying detection with respect to the complete post is our major motivation for this work. In addition to this, we need a fine-grained approach to detect hidden cyberbullying posts and posts containing intended typos in cyberbullying messages. In addition to this, our system must also have an elaborate intervention mechanism, where the reaction to a cyberbullying incident should be proportional to the severity of the cyberbullying attack.

Figure 1 leads us to make the inference that there can be two scenarios of cyberbullying incidents, discussed below.

C1: Cyberbullying from the Victim's perspective. A victim is a user at whom the cyberbullying attacks are directed by perpetrators. This can occur when the victim reads harmful content that are directed at them on their OSNs, as shown in Figure 1. In this case, we need a system that can assess the incoming feed of textual contents that are being posted onto the victim's online space.

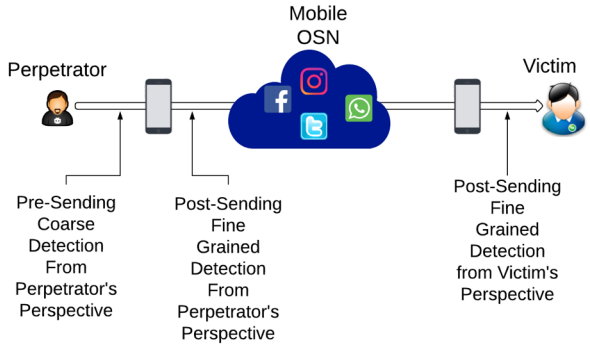

Figure 1. Detection Types of Cyberbullying in Mobile OSNs

C2: Cyberbullying from the Perpetrator's perspective. A perpetrator is a user who is posting cyberbullying textual content directed at a victim(s) on the OSN. A perpetrator would use a mobile device's keyboard to post cyberbullying content, as shown in Figure 1. In this case, we need a system that can assess the text that is being typed onto the OSN application running on the smartphone.

As mentioned in [6], a mobile cyberbullying system should be able to warn a perpetrator, characterized by C2, before the perpetrator can post a harmful message. Since this detection should be made before a message is sent, it should be quick, but at the same time should not be too restrictive. Research in [6] and [2] proves that an initial deterrent is very effective in preventing cyberbullying, achieving a 93% change in intention. But in addition to the initial quick detection, we also need a more fine-grained detection which can assess text messages for the detection of subtler forms of cyberbullying and a context-aware mechanism for detecting cyberbullying in posts, by taking the complete post story into consideration along with the newly added comments. In order to perform a fine-grained and context-aware analysis, the system must use machine learning techniques for deeper analysis of textual information in mobile devices. Hence, we need to detect cyberbullying effectively before and after posting messages, considering the above features of cyberbullying in mobile OSNs.

In addition to efficient detection and prediction components, a mobile cyberbullying system should also have an efficient intervention system that would take action based on the severity of the cyberbullying incident. Minor incidents of cyberbullying can be intervened by issuing warning messages/notifications to the users or by sending emails to concerned authorities, whereas more severe incidents might need more drastic intervention techniques, such as blurring the screen or blocking input to the OSN app.

3. OUR APPROACH

In order to achieve and prove that such a system with all the above mentioned features can be developed, we design a mobile cyberbullying defense system, *MCDefender*. *MCDefender* system can be described as having three core components – *Pre-Sending Quick Detection (PQD) Module, Post-Sending Fine-Grained Detection (PFD) Module and Dynamic Intervention Module*. The *MCDefender* framework overview is demonstrated in Figure 2. The components of our system can be classified into two main categories: System Components and Learning Components.

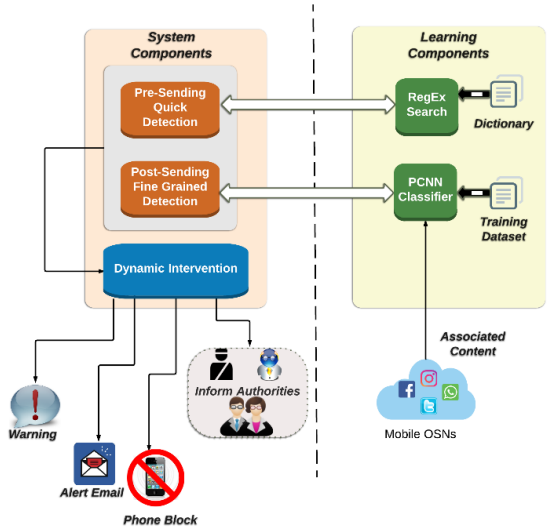

Figure 2. MCDefender Overview

The System Components are the core components that are responsible for cyberbullying detection and prediction. The PQD model warns the user before the user can make a cyberbullying attack. The PQD model uses keyword searching with regular expressions technique to detect cyberbullying posts before the posts are sent, using a dictionary of regular expressions (Figure 2, *RegEx search*) of high frequency bullying keywords. As demonstrated in Figure 2, if a cyberbullying keyword is matched with a regular expression, a dynamic intervention strategy is adopted. For an in-depth analysis using deep learning methods, *MCDefender* system uses the PFD component. This component uses the Pronunciation based CNN (PCNN) classifier for a deep analysis and detection of textual cyberbullying relevant to noisy data found in mobile OSNs. The Dynamic Intervention component gets triggered when cyberbullying is detected in a mobile device. The action is taken according to the severity of the cyberbullying attack. As seen in Figure 2, this can range from warning messages to blocking the users from using the OSN app on mobile devices.

3.1 Pre-Sending Quick Detection (PQD)

The PQD module of *MCDefender* system focuses on providing a quick deterrent/warning to the perpetrator before posting a cyberbullying text. In PQD module, we first pre-process the user input text by discarding unnecessary characters and symbols. The filtered text is then provided as input into a Regular Expressions search engine. A regular expression (regex) is a mechanism to describe a search pattern in a string. The efficiency of regular expressions in matching noisy data is demonstrated in [17]. In short, regexes are highly efficient in matching misspelt words. We have developed the *MCDefender* search engine as a RegEx search engine that uses a dictionary of high frequency cyberbullying regular expressions strings to quickly determine a potential cyberbullying attack. From [17], we can see that using RegEx for keyword matching is especially applicable for our system.

A problem with the PQD module is that it would detect non cyberbullying messages with high frequency cyberbullying words as cyberbullying messages. Due to this, the PQD module only provides a warning to the user of *probable* cyberbullying. The PFD module, discussed in the next section, performs a more in-depth analysis of messages. The PFD module also detects messages that do not contain cyberbullying words but may actually be of cyberbullying nature.

3.2 Post-Sending Fine-Grained Detection (PFD)

The PFD module of *MCDefender* focuses on detecting subtle forms of cyberbullying using deep learning techniques. Cyberbullying keywords are often also used in a friendly manner, where the intent is not to bully others. At the same time, a message can contain no high frequency cyberbullying words, but may still be cyberbullying, for example:

"Aw Adele gave birth to a baby :) is it fat and handicapped lol just murder it already lol"

The above post, which was in fact directed at a celebrity, demonstrates how a post having no cyberbullying words can in fact be a vicious case of cyberbullying. In order to predict cyberbullying in such posts, we have used deep learning techniques to develop a cyberbullying classifier. Convolutional Neural Networks (CNNs), which have been traditionally applied to image processing tasks, have proved to perform very well in natural language processing tasks. Hence, we have used CNNs as the neural network architecture in our classifier. We have used a Pronunciation based CNN (PCNN) technique to improve the precision of textual cyberbullying detection in noisy data [22]. We have used datasets from [11] which has been compiled from Twitter, for training our classifier. The dataset had noisy data, which was manually cleaned by us. Non alpha-numeric characters except apostrophe were removed. Then, a term compression operation was performed to ensure that no more than two consecutive occurrences of any character exists in a word. For example, "coolll" was replaced with "cooll", "bitchhh" was replaced with "bitchh" and so on. After the pre-processing operations, we used e-speak, which is an open source speech synthesizer software [12] to create phonetic representation of each word in our dataset. E-speak uses pronunciation rules and dictionary lookups to perform this conversion. We used ASCII codes for encoding the phoneme strings. We observed that most of the textual posts containing cyberbullying contained misspelled words. The word to pronunciation conversion helped us to map the misspelled words to the correctly pronounced words.

After the pronunciation based pre-processing of our dataset, the phonetic representation of each word was converted to a randomly initialized 300-dimensional vector. A zero vector was used to pad each sentence so that same length vectors can be presented to the PCNN network. Convolutional neural networks (CNN), originally created for image processing, have performed excellently in natural language processing (NLP), especially in sentiment analysis and question classification [7, 8, 14]. Inspired by their powerful feature representation capability, flexible structure, and high efficiency for training using a GPU, we adopted CNN as the baseline classifier. To have a clear performance comparison between PCNN and the baseline CNN, we used the same model architecture in [7]. As shown in the PCNN architecture diagram (Figure 3), only one layer of convolution and max-pooling was used with three different filter sizes. The sizes of the three convolutional filters were chosen to be 1, 2, and 3, slightly differing from the original paper. The filter sizes were chosen based on how many consecutive words are necessary to detect bullying content. The convolutional operation on m consecutive words is given in (1).

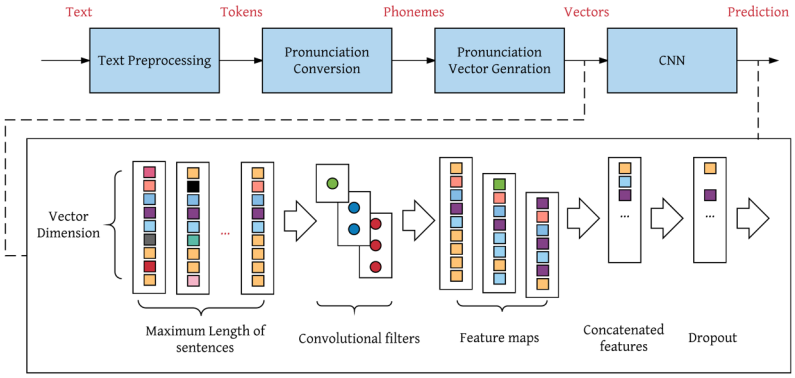

Figure 3. PCNN Architecture

$$hi = f(wcxi{:}i+m + bc) \quad (1)$$

Here, *xi:i+m, hi, wc, bc,* and *f* are the embedding matrix of m words, the feature generated by the operation, the weight and bias of the corresponding convolutional filter, and the activation function, respectively. We used the linear rectifier unit as the activation function. A max-pooling operation was applied to all the features from one convolutional filter. Then, the features were concatenated into *h*, a feature vector with dimensions equal to the number of filters applied. A softmax layer with dropout was applied to the output of the pooling layer to predict the class probability, *P*, as follows:

$$P(Y=i \mid X, \theta) = softmaxi(wsh + bs) \quad (2)$$

Here *X, h, Y, ws, bs, i, θ* are the input embedding matrix, feature vector from the convolutional and pooling layer, class prediction, weights of the penultimate layer, corresponding bias, class number, and parameter set, respectively. We used two separate CNN to establish a baseline. For the first baseline CNN, called CNN Pre-trained, we used 300-dimensional word-embedding based on Google's word-to-vector to create the feature set. Randomly generated vectors were used to create the feature set for the second baseline CNN, called CNN Random. For PCNN, the phoneme codes were randomly initialized into vectors for the feature set. All the embedding for CNN and PCNN was updated during the training process based on stochastic gradient descent [15]. Our method and the structure of the convolutional neural network are shown in Figure 3. The structure of PCNN is a cascaded combination of four components, as shown in the figure 3 below. Text messages are fed to text preprocessing module to extract meaningful tokens and discard nonsense and symbols. Then the tokens are converted into phonemes by a software called Espeak [2]. For example, "cool"and "cooll" share the same phoneme representation, "k'u:l".To leverage the CNN model, each phoneme representation is mapped to a vector according to a dictionary trained before. The architecture of the CNN model is similar to [1] but with only slight difference on the filter sizes. To be specific, the smallest size is set to be 1 to prevent losing strong features of some key words. Also using the similar architecture with [1] can make clearer the contribution of the pronunciation conversion. Finally, the prediction can be made by the last module, CNN.

4. DYNAMIC INTERVENTION

Dynamic intervention in *MCDefender* system is the action that is taken when a cyberbullying incident is detected. From Figure 1, we can see that there are three events that trigger the Intervention Module (Section 2). Hence, we have different intervention actions in accordance with the level at which cyberbullying is detected. *MCDefender* uses following intervention techniques:

Pre-Sending Quick Detection from Perpetrator's Perspective. This intervention scheme is issued when the PQD module detects a cyberbullying activity, before the perpetrator sends a cyberbullying post. In this case, warning messages, in the form of text notifications are issued to the potential perpetrator. The potential perpetrator is allowed to discard the warning and continue to send messages.

Post-Sending Fine-Grained and Context-Aware Detection from Perpetrator's Perspective. If a potential perpetrator discards warning from above step, a fine-grained analysis is carried out using the PFD module. If a cyberbullying activity is detected in the PFD module, a stricter action is intervened, for example, sending emails to authorities.

Post-Sending Fine-Grained and Context-Aware Detection from Victim's Perspective. In this intervention scheme the incoming feed of posts on a potential victim's mobile device are analyzed through the PFD module. If a cyberbullying activity is detected, we issue notifications to the victim's registered parents and other authorities. We also keep a count of such activities on the victim's mobile device. If such activities continue to occur, a more severe intervention technique of blurring the victim's social media app is adopted, so that no further harm can be inflicted on the victim.

5. IMPLEMENTATION AND EVALUATION

5.1 Prototype Implementation

We have implemented our two level cyberbullying detection architecture as an Android application. For cyberbullying detection from the perpetrator's perspective, we listen to keystrokes for posts directed at the Facebook app. We first run the entered text through the PQD module (Section 4.1). If we find a match, we immediately issue a warning message, in the form of a notification to the perpetrator. This acts as an immediate deterrent to the perpetrator, in case there indeed is a case of cyberbullying. If the perpetrator wishes to change her/his mind and alter the text, the new text that is entered is also evaluated in the same way. However, if the perpetrator sends the text without alterations, we begin analyses using the PFD module (Section 4.2), which performs fine-grained detection.

We have implemented our PCNN classifier using Google TensorFlow framework. For detection from the victim's perspective, we use the same Android Accessibility service to get the text that is embedded into Facebook views. In case cyberbullying is detected, we issue intervention procedures based on the severity of the case. For our dynamic intervention scheme, at the first level, we use warning messages to the user. For more severe cases, emails and messages are issued to the concerned authorities. If the case is extreme, we block the use of Facebook app by creating an overlay over the Facebook app.

5.2 Evaluation and Experiments

5.2.1 Evaluation of Detection Coverage

We have evaluated our approach for cyberbullying detection by comparing our solution with existing cyberbullying detection systems. From our observations, *ReThink* system [6] seems to be the most popular existing detection system. We have evaluated *MCDefender* based on the following metrics: *detection extent, detection effectiveness* and *intervention methods.*

ReThink does not attempt to detect the second form of cyberbullying, which is from the perspective of the victim. Thus, we conclude that the *detection extent* of R*eThink* is limited only if a perpetrator is using the *ReThink* system. *MCDefender* consists of detection components that also detect cyberbullying from a victim's perspective (Section 2). Thus, the *detection extent* of *MCDefender* system is larger when compared to the *detection extent* of *ReThink* system.

The *ReThink* system employs string matching to detect cyberbullying keywords. If a match is found, a warning is issued to the user *before* the message is sent to the social media servers. Simple string matching techniques are not very effective in detecting subtle forms of cyberbullying, nor are they efficient enough to detect misspelled words. Thus, the *detection effectiveness* of the *ReThink* system is very low. Our solution employs a Two Level detection strategy to detect cyberbullying in mobile systems. The first level detects cyberbullying that is being inflicted from the bully's mobile device, using regular expressions string queries. The second level of detection is made by the Fine-Grained Detection system.

In *ReThink*, the intervention strategy consists of warning messages that are issued when cyberbullying words are detected from a user input. We observed that the messages that are issued are repeated randomly and are generic in nature. This is not an effective strategy as action must be according to the severity of the cyberbullying attack. In our solution, if the severity of the attack is high, our solution can blur the OSN app interface so that harm cannot be further inflicted. This strategy is more dynamic than the strategy of *ReThink* as the intervention is proportional to the severity of cyberbullying. Hence we conclude that the *intervention methods* of our solution score better than that of *ReThink*.

5.2.2 MCDefender Performance

We use *time metric* for measuring the performance of our system as an Android app. Time metric is defined as the time in milliseconds that is taken by one operation. Figure 4 shows the plot of the word count Vs. time taken in milliseconds for the PFD and PQD modules, respectively. For this experiment, we took sentences from our labeled Twitter [11] corpus. We use sentences of a varying word count to perform our experiments by plotting the different word counts against the time taken as demonstrated in Figure 4. For plotting the outcome of our exeriments, the

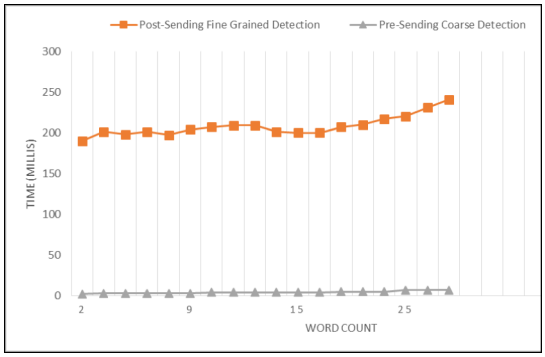

Figure 4. Time Performance

sentences has been arranged in ascending order of word count. We then plotted the ordered sentences against the time taken to analyze each sample by both the PFD and the PQD modules.

From Figure 4, we can observe that the PQD module is extremely fast for short to mid range sentences, with a substantial increase in computation time only for very large sentences. Since *MCDefender* is a mobile system, it is highly unlikely that a user would type sentenves that are as high as containing ~400 words [21]. This observation verifies our claim of using PQD module as a quick deterrent cybebullying detection method. From Figure 4, it is also clear that the PFD module takes comparitively higher time to generate prediction, as this process performs complex computation. But at the same time, we observe there is minimal time difference in computation time with the increase in word count. This can be attributed to the fact that the input vector length to the neural network architecture is same for all samples. The higher time requirements verifies our claim that the PFD module is to be used for post-sending of the potential cyberbullying messages, as such large delays will cause usability issues from the user's perspective. The lack of variation in time with the PFD module makes it suitable for the detection of cybebullying in a complete post.

5.2.3 PCNN and Regex Performance

We used precision, recall, accuracy, and F-1 score as metrics to evaluate the performance of our models. All of these metrics are based on the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The formulae for calculating the metrics are as follows:

$$Recall = TP / (TP + FN) \quad (5)$$

$$Precision = TP / (TP + FP) \quad (6)$$

Model	Precision	Recall	Accuracy	F1 score
PCNN	0.991	0.970	0.989	0.980
RegEx Search	-	-	0.927	-

Table 2. Comparison of Methods Used on the Twitter Dataset

$$F1 = 2 * Recall * Precision / (Recall + Precision) \quad (7)$$

We used ten-fold cross-validation on the Twitter dataset in order to remain consistent with the original authors [11]. Table 2 summarizes results of the PCNN and RegEx approaches tested on the Twitter dataset. In this experiment, PCNN performs extremely

well in all metrics, demonstrating the benefit of using the word to pronunciation conversion. Table 2 shows the accuracy of our RegEx technique, used in the PQD module for quick detection. We find that the RegEx model actually performs with quite high accuracy. Considering the performance of RegeEx on the Twitter dataset [11], we observed an accuracy of 92.7%. We note that even under noisy conditions, RegEx performs well from a quick detection perspective.

6. RELATED WORK

Kontostathis et al. [3] demonstrated the use of Bag of Words model on cyberbullying corpuses to achieve a precision of 91.25% on average. Lempa et al [2], showcased an Android application, embedded with two methods, to implement and evaluate cyberbullying detection as a mobile system. The first method used in this research is based on a brute force string matching algorithm that matches the user entered text with a dictionary of cyberbullying words and phrases. The other method extracts words and phrases as seed words and detects cyberbullying online with keyword categorization and relevance matching. The top precision of both methods reaches 89% and 91%, respectively [2]. Hosseinmardi et al. [4] used an Instagram dataset to detect cyberbullying on Instagram, which is a popular mobile social media application used for sharing photos. They defined two types of attacks – cyberaggression and cyberbullying and attempted to detect the two attacks separately. Cyberbullying is limited to comments in this case. Using a linear SVM model, they reported a maximum accuracy of 87% for detecting cyberbullying in Instagram. A simple solution to cyberbullying detection systems was made by the *ReThink* app [6]. *ReThink* is a smartphone application that warns users who are attempting to bully others using smartphone OSN apps by giving a warning message before the message is sent. This method has been known to be very effective to act as a deterrent to bullying [16] and hence the app claims that adolescents trying to bully other users change their mind 93% of the time as a result of a warning issued by the app.

7. CONCLUSION

In this work, we have proposed a novel system for defending against cyberbullying in mobile social networks. We have introduced a *two-level* cyberbullying detection method to effectively detect cyberbullying caused by mobile devices. In addition, we have proposed a set of intervention techniques that are in accordance with the severity of the cyberbullying attack. Our *two-level detection* mechanism consists of a Pre-Send detection scheme that acts as a deterrent to the bully and a fine-grained detection scheme that uses pronunciation-based CNN classifier to detect subtle forms of cyberbullying attacks. In the future, we plan to expand our system to detect *visual cyberbullying*, *which* is a new type of cyberbullying where perpetrators use images of/relating to their victims to harass or bully the victims [20].

8. ACKNOWLEDGEMENT

This work was partially supported by grants from National Science Foundation (NSF-CNS-1537924 and NSF-IIS-1527421).

9. REFERENCES

[1] Sameer Hinduja and Justin W. Patchin, (Springer Science+Business Media New York 2013). *Social Influences on Cyberbullying Behaviors Among Middle and High School Students.*

[2] P. Lempa, M. Ptaszynski, and F. Masui, "Cyberbullying Blocker Application for Android," presented at the 7th Language & Technology Conference (LTC'15), Poznan, Poland, 2015.

[3] A. Kontostathis, K. Reynolds, A. Garron, and L. Edwards, "Detecting cyberbullying: query terms and techniques," in *Proceedings of the 5th annual acm web science conference*, 2013, pp. 195-204.

[4] Hosseinmardi, H., Mattson, S.A., Rafiq, R.I., Han, R., Lv, Q. and Mishra, S., 2015. Detection of cyberbullying incidents on the instagram social network. arXiv preprint arXiv:1503.03909.

[5] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of Textual Cyberbullying," *The Social Mobile Web,* vol. 11, p. 02, 2011.

[6] Prabhu, Trisha N, *"Method to stop cyber-bullying before it occurs"*, United States Patent Application 20150365366, Kind Code: A1.

[7] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882,* 2014.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, pp. 2278-2324, 1998.

[9] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," in *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, 2011, pp. 241-244.

[10] Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman and Rosalind Picard, *"Common Sense Reasoning for Detection, Prevention,and Mitigation of Cyberbullying",* ACM Transactions on Interactive Intelligent Systems, Vol. 2, No. 3, Article 18, Publication date: September 2012.

[11] A. S. Kasture, "A predictive model to detect online cyberbullying," Auckland University of Technology, 2015.

[12] *eSpeak*. Available: http://espeak.sourceforge.net/

[13] W.-t. Yih, X. He, and C. Meek, "Semantic Parsing for Single-Relation Question Answering," in *ACL (2)*, 2014, pp. 643-648.

[14] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701,* 2012.

[15] Fumito Masui, Michal Ptaszynski and Nitta Taisei, *"An Apparatus and Method for Detection of Harmful Entries on Internet"*. Patent Application No. 2013-245813.

[16] Samaritan's Radar, http://www.samaritans.org/

[17] Pete Burnap, Walter Colombo, and Jonathan Scourfield. 2015. *"Machine classification and analysis of suicide-related communication on twitter"*. In Proceedings of the 26th ACM Conference on Hypertext & Social Media. ACM, 75–84.

[18] Lenhart, Amanda, Pew Research Center, April 2015, "Teen, Social Media and Technology Overview 2015".

[19] Hinduja, Sameer, and Justin W. Patchin. Bullying beyond the schoolyard: Preventing and responding to cyberbullying. Corwin Press, 2014.

[20] Zhong, H., Li, H., Squicciarini, A., Rajtmajer, S., Griffin, C., Miller, D., & Caragea, C. "Content-Driven Detection of Cyberbullying on the Instagram Social Network". in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, 2016.

[21] Lyddy, Fiona, et al. "An Analysis of Language in University Students' Text Messages." Journal of Computer- mediated Communication 19.3 (2014): 546-561.

[22] Xiang Zhang, Jonathan Tong, Nishant Vishwamitra, Elizabeth Whittaker, Joseph P. Mazer, Robin Kowalski, Hongxin Hu, Feng Luo, Jamie Macbeth and Edward Dillon. "Cyberbullying Detection with a Pronunciation Based Convolutional Neural Network" *ICMLA*. 2016.