

# Deep Learning Algorithm for Cyberbullying Detection

Monirah Abdullah Al-Ajlan

College of Computer and Information Sciences  
King Saud University  
Riyadh, Saudi Arabia

Mourad Ykhlef

College of Computer and Information Sciences  
King Saud University  
Riyadh, Saudi Arabia

**Abstract**—Cyberbullying is a crime where one person becomes the target of harassment and hate. Many cyberbullying detection approaches have been introduced, however, they were largely based on textual and user features. Most of the research found in the literature aimed at improving detection through introducing new features. However, as the number of features increases, the feature extraction and selection phases have become harder. On the other hand, no study has examined the meaning of words and semantics in cyberbullying. In order to bridge this gap, we propose a novel algorithms *CNN-CB* that eliminate the need for feature engineering and produce better prediction than traditional cyberbullying detection approaches. The proposed algorithm adapts the concept of word embedding where similar words have similar embedding. Therefore, bullying tweets will have similar representations and this will advance the detection. *CNN-CB* is based on convolutional neural network (CNN) and incorporates semantics through the use of word embedding. Experiments showed that *CNN-CB* algorithm outperform traditional content-based cyberbullying detection with an accuracy of 95%.

**Keywords**—Cyberbullying; convolutional neural network; CNN; detection; deep learning

## I. INTRODUCTION

With the proliferation of the internet and its anonymity nature, many ethical issues have emerged. Cyberbullying is among the most widely acknowledged problems by individuals and communities. It is defined as any violent, intentional action conducted by individuals or groups, using online channels repeatedly against a victim who does not have the potential to react [1]. Even though bullying has always been a critical issue and received much attention; the internet along with social media has only made the issue more critical and wide spread. This is because they open doors for predators and give them a 24/7 access to victims from all ages and backgrounds while keeping their identities anonymous [2]. For all the danger imposed by cyberbullying on victims and communities, this field of study is maturing, with a wealth of research and findings evolving every day. The vast range of existing cyberbullying studies are spanning fields like psychology, linguistics and computer science.

Psychologists recognized cyberbullying as being a phenomenon closely related to the well being of individuals. A study found in [3] where a total of 7000 students were examined, concluded that bullying contributes to higher levels of loneliness and lower levels of social well-being . Many

psychologists were asked in [4] about the appropriate actions that need to be taken in response to the growing number of cyberbullying incidents and they were in favour of the automatic monitoring of cyberbullying.

Automatic monitoring of cyberbullying has gained considerable interest in the computer science field. The aim has been to develop efficient mechanisms that mitigate cyberbullying incidents. Most of the literature considered it to be a binary classification task, where text is classified as bullying or non bullying [5]. This is achieved through extracting features from text and feeding them to a classification algorithm. Many studies have addressed cyberbullying detection from different perspective, however, all falls under four features categories: content-based, user-based, emotion-based and social-network based features.

Even though the state of art in cyberbullying detection is rapidly evolving, there are many problems that has arisen. A fundamental issue still present is that most research attempt to improve the detection process by suggesting new features. However, this approach might generate huge number of features that require careful feature extraction and selection phases which lead to computational overhead. Moreover, features are not always easy to be extracted. In fact, features can be easily fabricated [6]. Another drawback is that they fail to adapt to the changing nature of language. Offensive words that are considered features in most detection approaches are not static and change over time. As a result, detection approaches must not rely on static features rather on more automated mechanisms. Despite the success of current approaches, a core problem has not been addressed. The semantic of words, their meaning and relations have been overlooked.

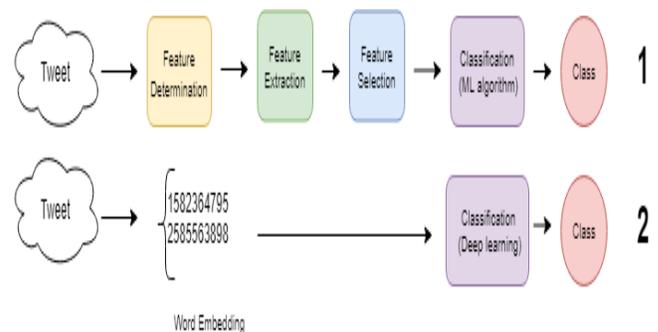


Fig. 1. Classical Machine Learning (1) vs CNN-CB (2).

In this article, we propose a convolutional neural network cyberbullying detection (*CNN-CB*) algorithm, which remedy the current unsolved problems. The primary goal is to develop an efficient detection approach capable of dealing with semantics and meaning and produces accurate result while keeping computational time and cost to a minimum. *CNN-CB* is based on deep learning which was on of MIT 10 Breakthrough Technologies Review in the year 2017 and 2013[7]. It is built upon the concept of convolutional neural network (CNN) which showed great success when applied to many classification tasks [8] [9] [10]. The most remarkable contribution is that *CNN-CB* is a cyberbullying detection algorithm that has shorten the classical detection workflow; it makes detections without any features. It transforms text into word embeddings and feeds them to a CNN. Previously, detection always started with feature extraction followed by, feature selection. Interestingly, *CNN-CB* has excluded these two steps and yet produced better result. Fig1 illustrates the traditional versus *CNN-CB* workflow.

This paper is organized as follows. Section II states the related work in cyberbullying. Then, section III describes *CNN-CB* in details. Section IV reports the experiments along with their results. Section V discusses the reported results. Finally, section VI concludes and summarizes this paper.

## II. RELATED WORK

Cyberbullying detection has a rapidly growing literature, even though researches addressing bullying are traced back to early 2010. The rich literature in this field can be divided into three categories: content-based, user-based and network based detection. Each category will be covered briefly, and comprehensive summary is presented in table I.

### A. Content-Based Detection

Among the first to tackle bullying in social media is [11], where a framework was built to incorporate Twitter streaming API for collecting tweets and then classifying them according to the content. Their work combined the essence of sentiment analysis and bullying detection. As a first phase, tweets are classified as being positive or negative and then they are further classified as positive containing bullying content, positive without bullying content, negative containing bullying content, and negative without bullying content. For the sake of classification, Naïve Bayes was implemented and resulted in a relatively high accuracy (70%). Another later research found in [12], incorporated statistical measures namely (TFIDF) and (LDA) along with topic models in order to extract relevance in documents. However, they did not rely on statistical measures only but extracted content features like: bad words and pronouns. Other researchers in [13], continued to pursue cyberbullying detection from content-based perspective however, they introduced new features like: emotions icon and dictionary of hieroglyphs. Their approach was tested using many learning algorithms: Naïve Bayes, SVM and J 48. And the best result was recorded with SVM achieving an accuracy of 81%. Another research [14],

presented a prototype system to be used by organization members to monitor social network sites and detect bullying incidents. The approach followed relied on recording bullying words and storing them in a database and then incorporate Twitter API to capture tweets and compare their content to the bullying material recorded earlier. Beside the promising innovative idea in their work, this prototype system has not been implemented yet.

### B. User-Based Detection

Many researchers believed that user information like age and number of tweets could indicate potentiality to harm others. In [15], researchers incorporated user information like number of tweets, number of followers and number of followings into the detection process. Their total features -user based and others- resulted in good predictions with an accuracy of 85%. Similarly, in [14] they added user age as feature along with a history of a user as a feature. They assume that if a user bullied in the past it is more likely for him to engage in bullying again. They investigated the effect of adding user features and concluded that it advances the recall with 5%. User-based features were also adopted in [16], where they added user gender and age to the feature set. The assumption was that different gender use different language and the people from different ages have different writing styles. Moreover, a new user feature was incorporated which was the user location.

### C. Network based Detection

An interesting perspective to cyberbullying detection studies the social structure of users. This starts by drawing network structure and deriving features from the graph. In [17], they focused on deriving features from social network graph. Features included: number of nodes indicating how large is the community and number of edges indicating how well connected is the community. Another research that addressed network based features is found in [12]. They used (Gephi) a graphical interface to visualise a user's connectivity based on the bullying posts. Then, they investigated the participants' role in the bullying, whether they are victims or predators.

## III. PROPOSED ALGORITHM

*CNN-CB* is an algorithm that advances current work in cyberbullying detection by adapting principles of deep learning instead of classical machine learning. *CNN-CB* architecture consists of four layers: embedding, convolutional, max pooling and dense which will be described in the following sub sections. The architecture and scope of every layer is shown in fig. 2. Its remarkable aspect is that it eliminates three classification phases previously employed by other detection algorithms, feature determination, extraction and selection. This is achieved through generating word embeddings (numerical vectors) for each word in a tweet and feeding them directly to a convolutional neural network. Detailed steps are explained in the following sub sections, and its pseudo code is represented in table II.

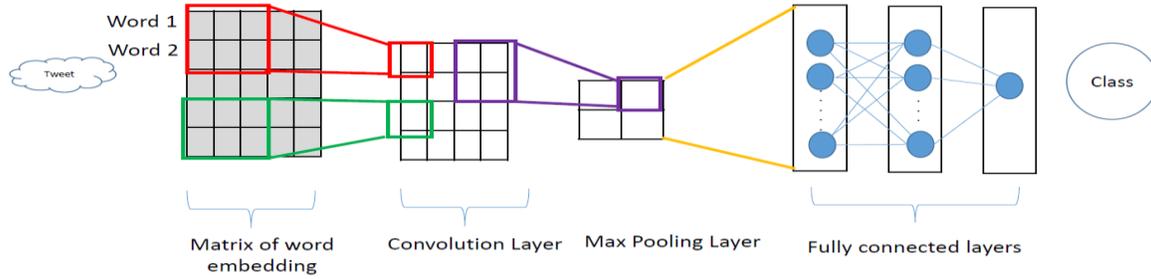


Fig. 2. CNN-CB Architecture.

TABLE I. CYBERBULLYING RESEARCH SUMMARY

Paper	Data Mining Task	Algorithm Used	Features
[15]	Classification	SVM, J48, Naïve Bayes	- Text - Profile - User graph
[18]	Classification	SVM.	- Text - Bag of words - Author
[12]	Classification	Lib SVM	- LDA /TFIDF - List of bad words - pronouns
[19]	Classification	J48, Naive Bayes, SMO, Bagging and Dagging.	- Social network - Text - Part of speech
[20]	Classification	SVM	- Text - Sentiment
[21]	Classification	C 4.5	- Text - List of bad words
[14]	Classification	SVM	- User - Content - Cyberbullying based
[22]	Classification	SVM, Naive Bayes	-Content -Profile
[16]	Classification Clustering	Fuzzy SVM	-User -Location -Text -Media
[23]	Classification	Fuzzy logic	-Text -Word statistics
[24]	Classification	SVM, J 48	-Text -TFIDF

A. Word Embedding

Word embeddings are a class of techniques used to generate numerical representation of textual material. A striking feature of word embedding is that they generate similar representations for semantically similar words. This remarkable feature enables a machine to actually understand what text means rather than dealing with it as strings of random numbers. In order to illustrate this great potential, fig 3 shows the similar words to word ‘smart’ along with their similarity score using word embedding provided by *Glove*. *Glove* [25] is one method of word embedding provided by

google. This works by collecting millions of words and training a neural network to learn the similarity or differences in meaning.

In the proposed *CNN-CB*, embedding layer provided by *Keras* [26] was adopted rather than pre trained embedding like *Glove*. What distinguish this specific choice of embedding (*keras*) is that it is task specific. In other words, it takes all text (cleaned tweets in this case) and generates a vector space of vocabulary. Thus, it is easier -both in time and resource-to compute. The use of word embedding made *CNN-CB*'s more advanced compared to traditional detection approaches since they incorporate semantics not just features extracted from raw text [27]. *Keras* embedding layer requires three parameters to be set prior to the construction of the vector space:

- Input dimension: specifies the total number of words in the vocabulary (whole corpus). This number is derived from the following. Let T be all tweets in the corpus.

$$T = \{t_1, t_2, t_3, \dots, t_n\}, n = \text{number of tweets}$$

$$\text{Input dimension} = \text{length}(\text{Tokenized}(T))$$

- Output dimension: specifies the size of the output vector from this layer.
- Input length: the length of each vector (maximum number of words per tweet). Twitter fixed maximum tweet length was not set, since this might change over time. Input length is calculated by using the following functions.

$$\text{Input length} = \max(\text{length for } t \text{ in } T)$$

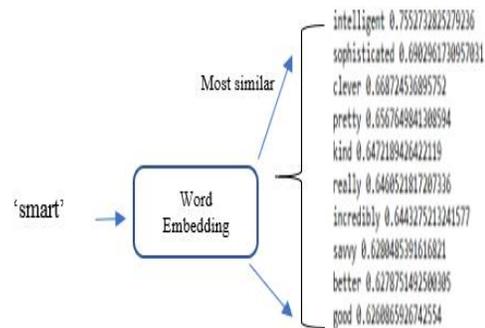


Fig. 3. Similar words to ‘smart’.

### B. Convolutional Layer

The second layer after the embedding layer (in case of text) is the convolutional layer. It is the heart of a convolutional neural network. Its task is to convolve around the input vector to detect features, therefore, it compresses the original input vector while preserving valuable features. This is achieved by creating a set of matrices called *filters* of random numbers called *weights*. Each filter is then independently convolved around the original input vector creating many feature maps through elementwise multiplication with the part of the input it is currently on [28]. In order to calculate the resulting feature map, Let  $V$  be the input vector of words, and  $F$  be the filter of size  $h*w$ , then the elementwise multiplication is calculated according to the following equation.

$$(V * F)_{xy} = \sum_{i=1}^h \sum_{j=1}^w V_{ij} \cdot F_{x+i-1, y+j-1} \quad (1)$$

### C. Max Pooling Layer

What distinguishes CNN and gives it robustness and ability to deal with complex data like image and large corpus, is that it compresses the input to smaller matrices. This remarkable ability is achieved by both convolutional and max pooling layers; thus, they are used after one another. Max pooling matrix simply slides across the output of a convolutional layer and finds the maximum value of the selected area. In this way, only meaningful and clear features are preserved.

### D. Dense Layer

All layers described so far were concerned with shaping data (tweets in our case) and compressing them in a meaningful way. So far, no classification has been done. This is exactly the job of dense layers. As in neural network, dense layers are set of fully connected layers [14]. In other words, each neuron is connected to all other neurons in the following layer. The number of dense layers varies, however, the last one must have 2 neurons corresponding to the number of classes in this case.

TABLE II. CNN BASED DETECTION ALGORITHM (CNN-CB)

CNN-CB	
Input:	List $T=t_1, t_2, t_3, \dots, t_n$ ( $n$ = number of tweets) Number $f$ ( $f$ =number of filters) Number $k$ ( $k$ =size of kernels) Number $p$ ( $p$ =pool size) Number count (count=number of classes) Number count ( $n$ =number of neurons)
Output:	Number $C=1$ or $0$ ( $0$ =no bullying, $1$ =bullying)
<b>Begin</b>	Tokenize all tweets <i>Token</i> Calculate total vocab, $vocab=length(Token)$ Calculate max tweet length, $len=max(length \text{ for } t \text{ in } T)$ Split tweets to testing <i>Test</i> , and Training <i>Train</i> Encode training tweets Encode testing tweets Create <i>Embedding</i> layer, Embedding( $vocab, len$ ) Create <i>Convolution</i> layer, Convolution( $f, k$ ) Create <i>Max Pooling</i> layer, Pooling( $p$ ) Create <i>Flatten</i> layer, Flatten() Create <i>Dense</i> layer, Dense( $n$ ) Create <i>Dense</i> layer, $C=Dense(count)$
<b>End</b>	

## IV. EVALUATION

Evaluation of the proposed algorithm aims to experimentally investigate crucial facts. First, that *CNN-CB* gives better results than traditional cyberbullying detection. Second, to evaluate other metrics like loss and recall. In order to have subjective evaluation, content-based detection was implemented for comparison with SVM algorithm. SVM was considered because a survey in [5], revealed that it is the mostly used in this domain. All experiments were run using *Windows* PC with 12 GB of *RAM*. All algorithms were programmed in *Python* [30] using *Spyder* environment [31]. *CNN-CB* was implemented using *Keras* [26] [29].

### A. Content-Based Detection (Cont)

There are many detection methods, however, a survey found in [5] stated that content-based methods are the most common with a total of 41 papers. Also, it has reported that SVM was the most common learning algorithm. The features included were: 1) the presence of bad words (bad words were retrieved from *noswearing.com* [32]); 2) the tweet's length;; 3) the presence of question marks since they indicate profane words; 4) the presence of exclamation marks since they indicate anger; 5) the presence of capital letters since they indicate anger.

### B. Dataset

The data set used in experiments were fetched from Twitter using Twitter streaming API [33]. A total of 39,000 tweets were retrieved from twitter public timeline. However, after annotating tweets, we found that there was an imbalanced class problem (very few bullying tweets). This has been solved by querying Twitter API with bad words from [32] so that it was more likely to return bullying tweets. After that, data were inspected and cleaned, removing duplicates and tweets with only pictures or URLs. A summary of the data collected for training and testing is presented in table III .

For data annotation, Figure 8 [34] human intelligence website was used. A job was posted, and sufficient instructions were given, and for quality purposes a test of 25 questions were required for a contributor to be accepted. Eventually, from those who succeeded the test with a percentage of 95%, two contributors were selected.

### C. Evaluation Metrics

Since cyberbullying detection is a classification task, the obvious choice of metric will be classification accuracy. However, this is an imbalanced class problem; so if we consider accuracy only as a metric then we might get an accuracy of 80% just by labelling all testing tweets with the majority class. This issue has been solved by considering two other metrics: recall and precision. All metrics are listed in the following equations.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$recall = \frac{TP}{TP+FN} \quad (3)$$

$$precision = \frac{TP}{TP+FP} \quad (4)$$

D. Result

In this section, comprehensive comparison between three cyberbullying detection approaches was conducted. The aim here is to prove that the proposed algorithm CNN-CB advances the current state of cyberbullying detection by providing better predictions (higher accuracy) although it eliminates the need for feature engineering. The series of experiments starts by testing CNN-CB with different values of filters, kernels, pooling and neurons to prove that changing values changes the quality of prediction. This experiment is reported in table IV. Moreover, further experiments are conducted to test the CNN-CB model. Fig.4 and fig. 5 shows the model accuracy and loss during every epoch respectively. The loss used in here is the mean squared error The third experiment shown in table V, is conducted with the traditional approach of cyberbullying detection, specifically content-based detection cont, and provides a summarized overview about its performance. Remarkably, the advancement of CNN-CB over traditional approach is clearly reported in fig.5, fig.6 and fig.7.

TABLE III. DATASET DISTRIBUTION

	Training	Testing
Bullying	9,000 tweets	2700 tweets
No-Bullying	21,000 tweets	6300 tweets

TABLE IV. CNN-CB BPERFORMANCE BY VARYING VALUES OF FILTERS, KERNELS, POOLING AND NEURONS

No. of filters	No. of kernels	Size of Pooling	No. of neurons	Accuracy (%)	Precision (%)	Recall (%)
4	5	1	4	66	99	70
4	5	1	16	95	77	82
<b>4</b>	<b>10</b>	<b>2</b>	<b>32</b>	<b>95</b>	<b>93</b>	<b>73</b>
4	10	2	43	95	86	77
4	10	2	54	94	80	80
10	20	2	4	65	99	70
10	20	4	16	93	81	79
10	20	4	32	93	83	79
10	25	4	54	94	81	80
10	25	4	64	93	79	81

TABLE V. CONTENT BASED CYBERBULLYING DETECTION PERFORMANCE

Algorithm	Accuracy (%)	Precision (%)	Recall (%)
SVM	81.32	73	70

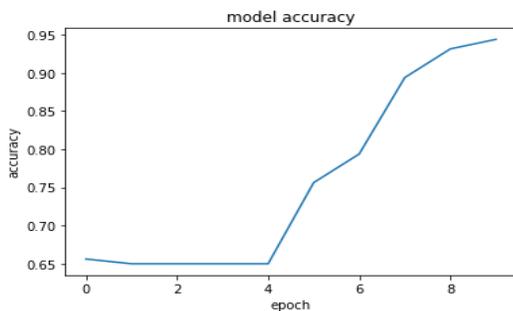


Fig. 4. Model Accuracy.

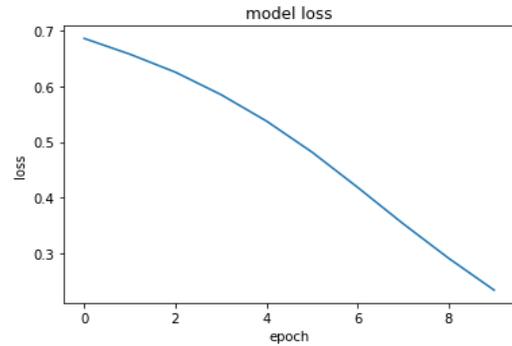


Fig. 5. Model Loss.

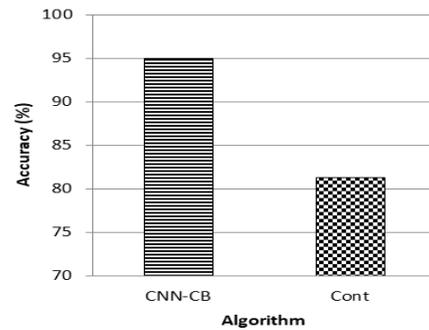


Fig. 6. CNN-CB vs Cont-SVM Accuracy.

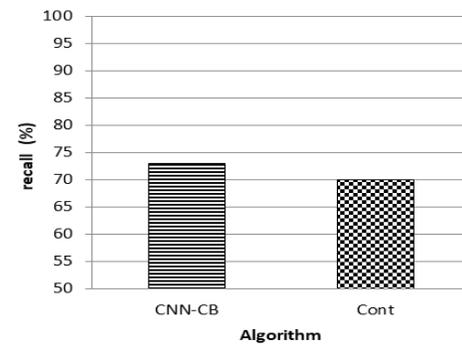


Fig. 7. CNN-CB vs Cont-SVM Recall.

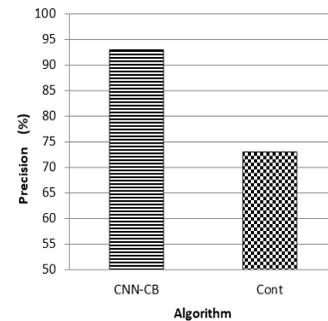


Fig. 8. CNN-CB vs Cont-SVM Precision

## V. DISCUSSION

Cyberbullying detection has been addressed in the literature with classical machine learning approaches, mainly content-based ones. However, the conducted experiments showed that *cont-SVM* gave an accuracy of 81%, like results reported by others in the literature.

The performance of *CNN-CB* during epochs, was always raising. This is because learning is evolving with every epoch. The model started with an accuracy of 65% but rose to 95% after 10 epochs. Model loss which represents a measurement of miss classifications also proved that increasing the number of epoch improve quality of predictions.

When *CNN-CB* is compared to traditional cyberbullying approach *cont-SVM*, *CNN-CB*, reported better results in the three metrics accuracy, precision and recall. This is true for all variations of parameters proving that feature engineering elimination did not degrade the performance but in fact there was a noticeable improvement of about 12% accuracy. Among the three studied metrics, accuracy shows the most noticeable difference. On the other hand, recall has slightly differed between the two algorithms with them being in the 70s.

It also has been evident from table IV that changing the CNN structure has a strong impact on the resulting accuracy. Some variations produced an accuracy of 66% whereas, some produced 95%.

## VI. CONCLUSION

Technology revolution advanced the quality of life, however, it gave predators a solid ground to conduct their harmful crimes. Internet crimes have become very dangerous since victims are targeted all the time and there are no chances for escape. Cyberbullying is one of the most critical internet crimes and research proved its critical consequences on victims. From suicide to lowering victims' self-esteem, cyberbullying control has been the focus of many psychological and technical research.

In this article, the issue of cyberbullying detection on Twitter has been tackled. The aim was to advance the current state of cyberbullying detection by shedding light on critical problems that have not been solved yet. To the best of our knowledge, there has been no research that considered eliminating features from the detection process and automating the process with a CNN. The proposed algorithm makes cyberbullying detection a fully automated process with no human expertise or involvement while guaranteeing better result. Comprehensive experiments proved that deep learning outperformed classical machine learning approaches in cyberbullying problem.

As for future work, we would like to adapt the proposed algorithms for Arabic content. Arabic language has different structure and rules so comprehensive Arabic natural language processing should be incorporated.

## REFERENCES

- [1] R. Shetgiri, "Bullying and Victimization Among Children," *Adv. Pediatr.*, vol. 60, no. 1, pp. 33–51, Jul. 2013.
- [2] R. Donegan, "Bullying and cyberbullying: History, statistics, law, prevention and analysis," *Elon J. Undergrad. Res. Commun.*, vol. 3, no. 1, pp. 33–42, 2012.
- [3] D. Halpern, M. Piña, and J. Vásquez, "Loneliness, personal and social well-being: towards a conceptualization of the effects of cyberbullying/Soledad, bienestar social e individual: hacia una conceptualización de los efectos del cyberbullying," *Cult. y Educ.*, vol. 29, no. 4, pp. 703–727, 2017.
- [4] K. Van Royen, K. Poels, W. Daelemans, and H. Vandebosch, "Telematics and Informatics Automatic monitoring of cyberbullying on social networking sites: From technological feasibility to desirability," *Telemat. INFORMATICS*, 2014.
- [5] S. Salawu, Y. He, and J. Lumsden, "Approaches to Automated Detection of Cyberbullying: A Survey," vol. 3045, no. c, pp. 1–20, 2017.
- [6] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," *Proc. Australas. Comput. Sci. Week Multiconference - ACSW '17*, pp. 1–8, 2017.
- [7] MIT Technology Review, "These are the 10 breakthrough technologies you need to know about right now," 2018. [Online]. Available: <https://www.technologyreview.com/lists/technologies/2017/>. [Accessed: 02-Mar-2018].
- [8] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *arXiv Prepr. arXiv1412.1058*, 2014.
- [9] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.
- [10] A. J. McMinn, Y. Moshfeghi, and J. M. Jose, "Building a large-scale corpus for evaluating event detection on twitter," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 409–418.
- [11] H. Sanchez and S. Kumar, "Twitter bullying detection," *ser. NSDI*, vol. 12, p. 15, 2011.
- [12] V. Nahar, X. Li, and C. Pang, "An effective approach for cyberbullying detection," *Commun. Inf. Sci. Manag. Eng.*, vol. 3, no. 5, p. 238, 2013.
- [13] C. Paper, "Methods for detection of cyberbullying: A survey," no. October, 2016.
- [14] M. Dadvar, D. Trieschnigg, R. Ordelman, and F. De Jong, "Improving cyberbullying detection with user context," pp. 2–5.
- [15] E. A. Abozinadah, A. V Mbaziira, and J. H. J. Jr, "Detection of Abusive Accounts with Arabic Tweets," vol. 1, no. 2, 2015.
- [16] V. Nahar, S. Al-Maskari, X. Li, and C. Pang, "Semi-supervised learning for cyberbullying detection in social networks," in *Australasian Database Conference*, 2014, pp. 160–171.
- [17] P. K. Atrey, "Cyber Bullying Detection Using Social and Textual Analysis," pp. 3–6, 2014.
- [18] S. O. Sood, E. F. Churchill, and J. Antin, "Automatic Identification of Personal Insults on Social News Sites," vol. 63, no. 4, pp. 270–285, 2012.
- [19] Q. Huang, V. K. Singh, and P. K. Atrey, "Cyber bullying detection using social and textual analysis," in *Proceedings of the 3rd International Workshop on Socially-Aware Multimedia*, 2014, pp. 3–6.
- [20] R. Sugandhi, "Automatic Monitoring and Prevention of Cyberbullying Automatic Monitoring and Prevention of Cyberbullying," no. June, 2016.
- [21] K. Reynolds, A. Kontostathis, and L. Edwards, "Using Machine Learning to Detect Cyberbullying."
- [22] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra, "Detection of Cyberbullying Incidents on the Instagram Social Network," 2014.
- [23] B. S. Nandhini and J. I. Sheeba, "Online Social Network Bullying Detection Using Intelligence Techniques," vol. 45, pp. 485–492, 2015.
- [24] K. Dinakar, B. Jones, C. Havasi, and H. Lieberman, "Common Sense Reasoning for Detection, Prevention, and Mitigation of Cyberbullying," vol. 2, no. 3, 2012.
- [25] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on*

- empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
- [26] F. Chollet, “Keras,” GitHub, 2015. [Online]. Available: <https://github.com/fchollet/keras>. [Accessed: 26-Aug-2018].
- [27] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *J. Mach. Learn. Res.*, vol. 3, no. Feb, pp. 1137–1155, 2003.
- [28] N. Majumder, A. Gelbukh, I. P. Nacional, and E. Cambria, “Deep learning-based document modeling for personality detection from text,” *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 74–79, 2017.
- [29] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. “O’Reilly Media, Inc.,” 2017.
- [30] “Python.” [Online]. Available: <http://www.python.org>. [Accessed: 28-Aug-2018].
- [31] “The Scientific Python Development Environment,” 2018. [Online]. Available: <https://www.spyder-ide.org/>. [Accessed: 28-Aug-2018].
- [32] “Swear Word List.” [Online]. Available: <https://www.noswearing.com/>. [Accessed: 28-Aug-2018].
- [33] “Python Twitter,” 2016. [Online]. Available: <https://python-twitter.readthedocs.io>. [Accessed: 28-Aug-2018].
- [34] Figure Eight Inc, “Figure Eight,” 2018. [Online]. Available: <https://www.figure-eight.com/>. [Accessed: 28-Aug-2018].