Cyberbullying Detection in Textual Modality

Evangeline D[1], Amy S Vadakkan[2], Sachin R S[3], Aakifha Khateeb[4], Bhaskar C[5]
Assistant Professor, Department of ISE, M S Ramaiah Institute of Technology, Bangalore, India[1]
Student, Department of ISE, M S Ramaiah Institute of Technology, Bangalore, India[2, 3, 4, 5]

***Abstract*—Cyberbullying is the use of technology to harass, threaten or target another individual. Online bullying can be particularly damaging and upsetting since it is usually anonymous and it's often hard to trace the bully. Sometimes cyberbullying can lead to issues like anxiety, depression, shame, suicide, etc. Most of the cyberbullying cases are not revealed to the public and the number of cases reported to the legal system is only few. Certain victims do not reveal their bully experiences out of shame or due to difficult procedures for reporting to the legal system. Our cyberbullying detection system aims to bring cases involving cyberbullying under control by detecting and warning the bully. Such cases are also reported to appropriate authorities, which can then be verified and necessary actions can be taken depending on the situation. The technology stack used for implementation include Flask, Scikit learn, Chat application APIs, Firebase, HTML, Javascript and CSS. The model was tested on classifiers like SVM, KNN, Logistic regression and Random Forest. F1 score was used as a metric to assess the four models. While analyzing the performances of these models, it was observed that Random Forest Classifier outperformed all the models. F1 score of 93.48% was achieved using the Random Forest Classifier.**

***Keywords*—Cyberbullying detection; support vector machine (SVM); kNN (k nearest neighbor); logistic regression; random forest classifier**

I. Introduction

The most common type of online bullying is mean comments which includes use of aggressive and pejorative words, threats, profile hacking etc. [11][12][13][14]. Nearly 8 out of 10 individuals are subject to the different types of cyberbullying in India. Out of these around 63% faced online abuses and insults, and 59% were subject to false rumours and gossip for degrading their image. 64% of victims receive an aggressive instant message when they are bullied. 7 in 10 young people experience cyberbullying before they hit the age of 18. About 37% of children between 12 and 17 years experienced cyberbullying at least once. One in four children fall victim to cyber bullies. In just one year, cyberbullying of teenagers and Indian women has increased by 36%. Only 4.6% of the cases are reported to the authorities and the rest go unnoticed or are hidden by victims to save themselves from further damage. Cases of cyber stalking or bullying of women or children increased by more than 36% from 2018 to 2020, data released recently by the National Crime Records Bureau showed.

Most of the comments that are posted on social media are noticed by people, but a large number of cases involving cyberbullying in messages are not shown in the public by victims to protect themselves from shame. These events can severly impact the one getting bullied and can sometimes lead to suicide. Currently studies and projects that are carried out in this area only include models to classify single sentences as a comment on bullying or not. What differentiates our model from existing ones is that we capture the message, the context and details of the bully and report it to authorities. With the use of our cyberbullying detection system, messages indicating cyberbullying can be detected and reported so that such events do not go unnoticed.

The rest of the paper is organized as follows: Section II summarizes contemporary research works carried out in cyberbullying detection. While Section III elaborates the methodology employed, Section IV discusses the results as an outcome of our work carried out. Section V concludes the findings and details the future work.

II. Literature Review

Many existing approaches were proposed in [1]. Authors have worked on the detection of Cyberbullying over comments posted on Instagram. Preprocessing techniques performed include unimportant character removal and removal of stop words. The machine learning model used was the Linear Support Vector Machine (LSVM). The metrics used for evaluation were accuracy, precision and recall. The model was designed only for detection of highly negative social media posts, more features and detailed labelling surveys can improve accuracy. In [2], authors have worked on identifying tweets related to cyberbullying by using PHP and HTML with MySQL and Twitter API. The preprocessing steps carried out were removal of punctuations and emotional icons. The detection of cyberbullying in tweets was done using a simple keyword search, in which each word present in the tweet was compared with the words in the dataset. The model did not consider the context of tweet, accuracy, informal language and abbreviations.

In [3], authors worked on the detection of cyberbullying on ASKfm which is a platform that allows users to ask and answer questions anonymously. Data cleaning steps done were removal of white spaces and replacement of abbreviations. The preprocessing steps consisted of tokenization, POS tagging and lemmatization. The machine learning model used was SVM (Support Vector Machine) and the evaluation metric considered was F1 score. The model only detected cyberbullying, it did not warn the bully or report the same to authorities. In [4], authors worked on identification and classification of Cyberbully Incidents using Bystander Intervention Model. The proposed model focused mainly on the analysis of direct intervention by bystanders. The dataset used consisted of posts and activities from facebook. Data preprocessing steps included were removal of whitespaces and

stop words. The machine learning model used for implementation was Random Forest Classifier. A limited dataset with limited accuracy was used for binary classification. In [5], the authors worked on the design of Semantic Framework for detecting Cyberbullying on social media. The main focus of the paper was detection of cyberbullying using Semantic learning. Removing stop words and Tokenization were the preprocessing steps performed. Sentiment Analysis was done on comments from posts. The model concentrated only on binary classification of comments and did not include reporting of comments related to cyberbullying to authorities. In [6], the authors' approach is validated on a dataset of over 3000 images along with peer-generated comments posted on the Instagram photo-sharing network, running comprehensive experiments using a variety of classifiers and feature sets. In this work, methods for detecting cyberbullying in commentaries following shared images on Instagram are developed. Classification of images and captions themselves are potential targets for cyberbullies. Standard k-fold validation technique is used to train data. It is a lengthy process. The training algorithm has to be rerun from scratch k times, which means it takes k times as much computation to make an evaluation. In [7], the paper proposes a supervised machine learning approach for detecting and preventing cyberbullying. Several classifiers are used to train and recognize bullying actions. The evaluation of the proposed approach on cyberbullying dataset shows that Neural Network performs better. Preprocessing is done, and then feature extraction is performed. The extracted features are fed into a classification algorithm to train, and test the classifier and hence use it in the prediction phase. Two classifiers, namely, SVM and Neural Network are used. This model of detecting cyberbullying patterns is limited by the size of training data. Thus, a larger amount of cyberbullying data is needed to improve the performance. In [8], authors worked on aggressive text detection for cyberbullying. The proposed model automatically maps a document with an aggressiveness score, thus treating aggressive text detection as a regression problem and explores different approaches for this purpose. These include lexicon-based, supervised, fuzzy, and statistical approaches. The different methods were tested over a dataset extracted from Twitter and compared them against human evaluation. The results favored approaches that considered several features particularly the presence of swear or profane words. The approaches used could be refined to better handle difficult cases, testing more supervised approaches using a larger dataset and building a framework for cyberbullying automatic identification. In [9], the authors have focused on detection and mitigation of cyberbullying in English and Arabic. The dataset was taken from Facebook posts and Twitter feeds. Their preprocessing involved removing tweets other than English and Arabic. Naive Bayes and Support Vector Machine were used to build the model. Recall and Precision were used as a metric for evaluating the built model. Enhancing the performance measures achieved by the system by using hybrid training models, such as combinations of Distance Functions, NB and SVM would make the model more effective in detecting cyberbullying. In [10], the authors have used methodologies to extract texts sent by the user and network based attributes are used to study the properties of bullies and aggressors. Dataset was taken from youtube comments and Twitter handles. The preprocessing includes replacing abbreviations with full phrases. Multiple models were used such as Logistic Regression, SVM and Gradient Boost. These models were only able to achieve an accuracy between 70 and 75% and it was only able to give a binary classification.

Some gaps were identified in the works discussed in this section. In [1], cyberbullying detection is limited only to English and Arabic. In [2][3][8], reporting to authorities is not done at all. In [6][7], only a limited dataset was employed. The works mentioned in [5][6] performs only with limited accuracy. Hence, in our paper, we have focused on working with large dataset and improving accuracy. We have also worked on reporting to authorities.

III. METHODOLOGY

The system consists of two main parts, a backend that captures messages on a chat application and calculates the probability that the sent message belongs to categories like toxic, severe toxic, obscene, insult or identity hate and a front end which displays messages that have a high probability of the message belonging to one of the six categories mentioned. The system is designed with the following objectives.

- Detection of cyberbullying in text messages.
- Reporting to appropriate authorities for initiating suitable action.

The technology stack used for backend are Flask, Scikit learn and Firebase. Flask is a micro web framework written in Python which contains third-party python libraries used for developing web applications. Flask was considered for this system since it was easy to integrate it with the machine learning model which was also written in python. Scikit Learn is one of the most useful libraries for machine learning in python and contains several tools and implementations of many machine learning algorithms. This has been used to determine the probabilities of messages being a bully message with the help of a random forest classifier.

Firebase which is a Backend-as-a-service provided by Google was chosen as the database to store all the details related to cyberbullying messages sent on the chat application. Email notifications are also triggered when database updates are made. The front end web application that displays the dashboard is built using HTML, CSS and Javascript. This dashboard is used by authorities to view details associated with the cyberbully messages.

A. Dataset

Toxic comment dataset was chosen for this purpose which was sourced from Wikipedia's talk page edits. This dataset was posted on Kaggle by the Conversation AI team, a research initiative founded by Jigsaw and Google. It essentially consists of 1,59,571 rows and 7 columns. Each row has a unique comment along with its comment id and the labels that it belongs to. Various labels that describe the comment in the dataset are - toxic, severe_toxic, obscene, threat, insult and identity_hate. Libraries used include numpy, re, panda, nltk, Matplotlib, wordcloud, sklearn, pickle.

B. Exploratory Analysis

Exploratory analysis is the process of analysing the dataset and displaying it in the form of graphs for better understanding of how the data is structured. A bar graph is plotted to see the unique comments that fall under different categories.

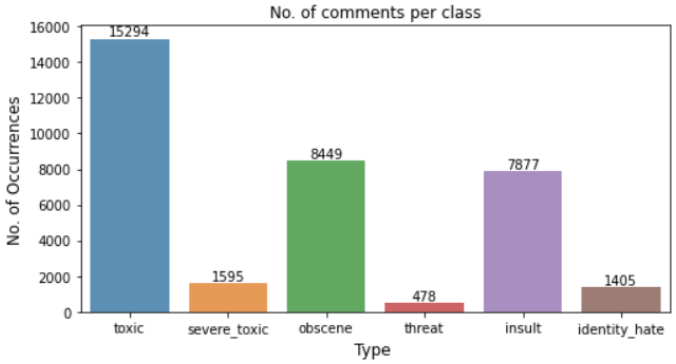

Fig. 1. Bar Chart Showing Number of Comments Per Class.

Fig. 1 shows that there are 15,294 toxic messages, 1595 severe_toxic messages 8449 obscene messages 478 threat messages, 7877 insult messages and 1405 identity_hate messages.

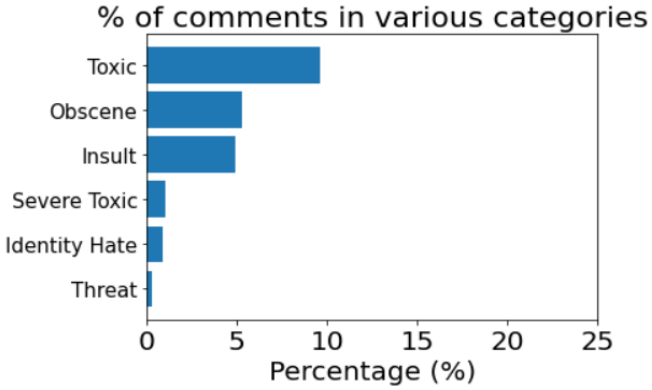

Fig. 2. Percentage of Comments in Various Categories.

Fig. 2 represents the percentage of comments that belong to a particular hate label. As we can see that only about 30% of comments fall under one of the given labels and a certain set of preprocessing has to be done in order to avoid class imbalance problems, see Fig. 3.

Fig. 3. Word Cloud.

C. Data Preprocessing

Data preprocessing is a technique that involves transformation of raw data into a format that can be utilized by machine learning models. The comments present in the dataset contain characters that are not essential in determining if a message is related to cyberbullying, hence such characters must be removed to increase the efficiency of the machine learning models. Various preprocessing steps have been carried out to clean the raw data. They include removal of numbers in combination with letters, replacing punctuations with white spaces, conversion of the comments to lower case, removal of escape sequences and replacing it with white spaces, removal of non ascii characters, removal of stop words, etc.

Since the data has lot of non-toxic comments and less toxic comments, training the model can make it prone to identifying more non-toxic comments rather than toxic comments. Hence, balancing of non-toxic and toxic comments has been done. After the basic preprocessing of the data, the dataset was separated into six sections. Each section contains the comments along with one of the categories in the dataset. Comments that were toxic or belonged to other forms of toxicity made up for less than 10% of the comments in the dataset. This led to a class imbalance problem where the class distribution was unequal or skewed. To overcome this issue, we took a subset of the data where the proportion of comments related to toxicity was at least 20% in relation to non-toxic comments. In order to make the subsets first the number of comments under each category was counted. If the category had 5000 rows, another 5000 rows of comments not belonging to the category were considered and the two were combined to form a new section. In case, a category did not have 5000 rows, then 20% of the comments belonging to the category and 80% of the comments not belonging to the category were chosen for the combination. In this manner six different sections or data frames were made per category.

Fig. 4 gives an overview of the data frames after eliminating the class imbalance problem. After the dataset was balanced, it was vectorized using Tfidf vectorizer which tokenizes the documents, learns the vocabulary given in the dataset and calculates the document frequency weighting which is essential in building a machine learning model. As machine learning models do not understand words or letters, they are converted to a machine understandable format using this method. 1-gram is used to split and tokenize the data, where it breaks the sentences into individual words and calculates the document frequency weight using Tfidf vectorizer.

	DF name	No. of pts (1)	No. of pts (0)	Total data pts
Toxic	data_tox_done	5000	5000	10000
Severe Toxic	data_sev_done	1595	1595	3190
Obscene (8449)	data_obs_done	5000	5000	10000
Threat (478)	data_thr_done	478	1912	2390
Insult (7877)	data_ins_done	5000	5000	10000
Identity Hate (1405)	data_ide_done	1405	5620	7025

Fig. 4. Overview of the Dataframes.

D. Backend

The backend was built using Flask, the database used was Firebase and Telegram API. SMTPLib (Simple Mail Transfer Protocol) module, which defines an smtp client session object is used to send emails.

When the server is run, the vectorizer and Random Forest model for each category is unpickling and loaded into the memory. We have used the Telegram API to capture messages sent by users. These messages are first vectorized and then passed to the machine learning model. These models return their respective names and the associated probabilities. The returned probabilities are compared with a threshold to identify bully messages. If the probability exceeds the threshold of 0.7, the model with the highest probability is chosen as the category that identifies the type of bullying. A warning is sent for the first bully message in the conversation, if more of such messages are sent further actions are taken.

A message queue is maintained to hold a few messages before and after the second bully message in the conversation. This queue is used to store the context of the messages being sent. The details of the messages and users such as username, chat id, time stamp, etc., are also stored in the message queue. Once the message queue is filled, the queue is pushed to the database along with the type of bullying and its probability. This event also triggers an email to the authority which alerts them about the update of the database so that they can review the conversation. This service is hosted on Heroku where the service is kept running so that messages can be monitored in real time.

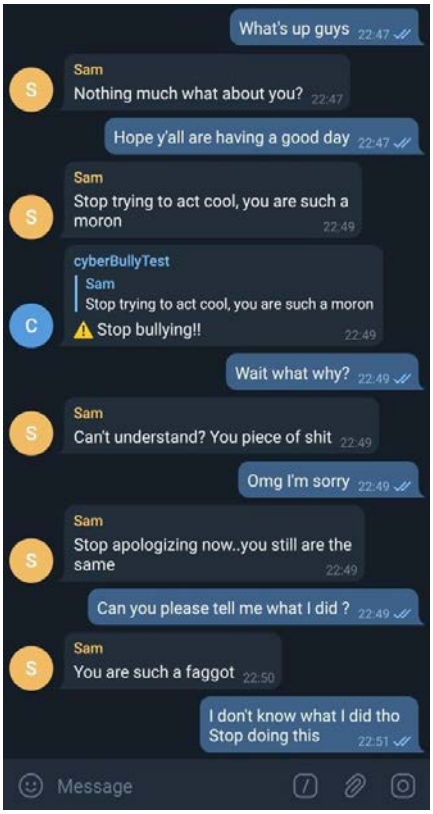

Fig. 5. Snapshot of a Cyberbullying Conversation.

Fig. 5 is a snapshot of a conversation that involves bullying. A warning is sent by the bot as a reply to the first bully message. If the user responsible for sending the message continues to do so by ignoring the warning, the messages are captured as evidence along with their details and these are saved to the database.

E. Frontend

The email that is sent to the authority notifies the authority when the database has been updated and it consists of the dashboard URL. On clicking the URL mentioned in the email for dashboard, the user lands on the login page where the person has to login using the official cyber cell credentials. Firebase authentication was used for this purpose.

Upon successful login, the authority is redirected to the pending page as shown in Fig. 6 which consists of all the conversations along with details like the type of toxicity, level of toxicity, name and user Id of the bully. The authority is given an option to either delete the conversations or approve them if they are legitimate. Approved messages are displayed under the approved tab.

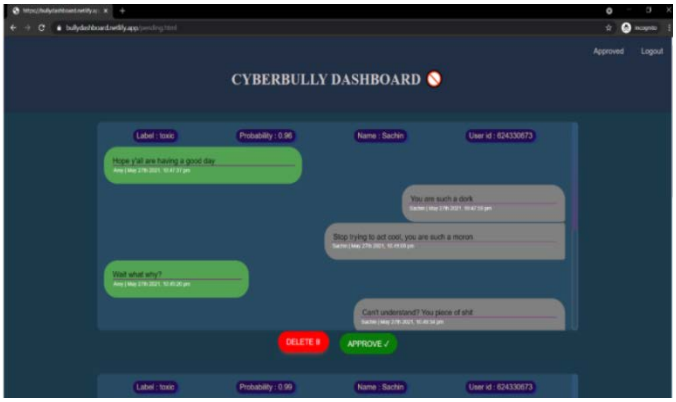

Fig. 6. Pending Reports Displayed in the Dashboard.

IV. Result and Discussion

The evaluation metric chosen was F1 score. F1 score was selected since it takes into consideration both False Positive and False Negative values, our goal was to minimise these values. A table comparing the F1 scores obtained using each of the machine learning models for the six data frames is shown below in Table I.

TABLE I. Comparison of F1 Score for the Models Used

Measure	Logistic Regression	KNN	SVM	Random Forest
F1 Score (Toxic)	0.861234	0.185120	0.876133	0.838055
F1 Score (severe_toxic)	0.927879	0.857416	0.926004	0.934874
F1 Score (obscene)	0.908655	0.519056	0.921378	0.909091
F1 Score (insult)	0.896599	0.257992	0.902619	0.883993
F1 Score (threat)	0.628821	0.720000	0.786765	0.795539
F1 Score (identity_hate)	0.699029	0.230159	0.797516	0.768448

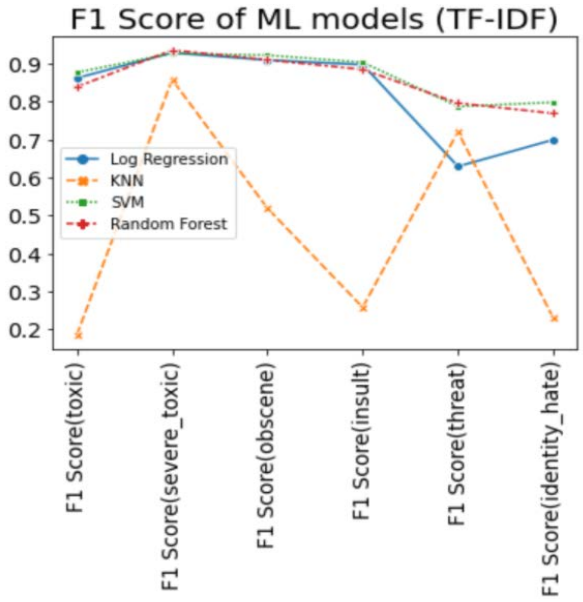

Fig. 7. F1 Score Comparison Plot.

By observing the values in Fig. 7, we see that SVM and Random Forest perform comparatively better than Logistic Regression and KNN classifiers. The plot in Fig. 7 shows that Random Forest represented by the red line and SVM represented by the green line perform better than the rest. So, Random Forest is chosen as the best performing model since results are returned in probabilities which were useful for our cyberbully detection system. Random Forest also makes use of multiple decision trees and hence gives a more accurate result.

V. Conclusion and Future Work

There are many people who are falling prey to cyberbullying, which goes unnoticed. It is high time that a system is made which helps in preventing these crimes. Our cyberbullying detection system aims to bring all these cases of cyberbullying under control by detecting and warning the bully. Then these cases are also reported to appropriate local authorities, which can be verified and required steps and actions can be taken depending upon the situation. Our model is built by using Chat application APIs, Firebase, Flask, Scikit learn, HTML, CSS and Javascript. It also gives high accuracy compared to the existing projects. It can be feasibly used and would show appropriate information whenever one is cyberbullied.

In future, CNN will be applied. Also, the algorithm will be applied on huge datasets and accuracy can be improved further. Certain issues like reducing false alarms, educating the users of the usability feature of cyberbullying detection and reporting to authorities, framing of privacy features of the platform and having moderators to review the conversations will be addressed in future work.

References

[1] Batoul Haidar, Maroun Chamoun & Ahmed Serhrouchni. (2017). A Multilingual System for Cyberbullying Detection: Arabic Content Detection using Machine Learning. *In : Proceedings of Advances in Science, Technology and Engineering Systems Journal.*

[2] Cynthia Van Hee, Jacobs G, Emmery C, Desmet B, Lefever E & Verhoeven B. (2018). Automatic detection of Cyberbullying in social media text. *In: Proceedings of PLOS One.*

[3] Gurpinder Singh , Vijay Dhir & Vijay Rana. Design of Semantic Framework for Detecting Cyberbullying on Social Media. *In: Proceedings of International Journal of Scientific Research and Reviews.*

[4] Haoti Zhong, Anna Squicciarini, Sarah Rajtmajer, Christopher Griffin, David Miller & Cornelia Caragea. (2016). Content-Driven Detection of Cyberbullying on the Instagram Social Network. *In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence.*

[5] Hosseinmardi, Sabrina Arredondo, Rahat Ibn Rafiq, Richard Han, Qin Lv & Shivakant Mishra. (2015). Detection of Cyberbullying on Instagram Social Network. *In: Proceedings of Association for the Advancement of Artificial Intelligence.*

[6] J.I. Sheeba, S. Pradeep Devaneyan & Revathy Cadiravane. (2019). Identification and Classification of Cyberbully Incidents using Bystander Intervention Model. *In: Proceedings of International Journal of Recent Technology and Engineering (IJRTE).*

[7] John Hani, Mohamed Nashaat, Mostafa Ahmed, Zeyad Emad, Eslam Amer & Ammar Mohammed. (2019). Social Media Cyberbullying Detection using Machine Learning. *In: Proceedings of (IJACSA) International Journal of Advanced Computer Science and Applications.*

[8] Kshitiz Sahay, Harsimran Singh Khaira, Prince Kukreja & Nishchay Shukla. (2018). Detecting Cyberbullying and Aggression in Social Commentary using NLP and Machine Learning. *In: Proceedings of* International Journal of Engineering Technology Science and Research.

[9] Laura P. Del Bosque & Sara Elena Garza. (2014). Aggressive Text Detection for Cyberbullying. In: Proceedings of Springer International Publishing, Switzerland.

[10] Liew Choong Hon & Kasturi Dewi Varathan. (2015). Cyberbullying detection system on Twitter. In: Proceedings of International Journal of Information Systems and Engineering, 1:1.

[11] Monirah Abdullah Al-Ajlan and Mourad Ykhlef, "Deep Learning Algorithm for Cyberbullying Detection" International Journal of Advanced Computer Science and Applications(IJACSA), 9(9), 2018. http://dx.doi.org/10.14569/IJACSA.2018.090927.

[12] Ximena M. Cuzcano and Victor H. Ayma, "A Comparison of Classification Models to Detect Cyberbullying in the Peruvian Spanish Language on Twitter" International Journal of Advanced Computer Science and Applications(IJACSA), 11(10), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0111018.

[13] Rolfy Nixon Montufar Mercado, Hernan Faustino Chacca Chuctaya and Eveling Gloria Castro Gutierrez, "Automatic Cyberbullying Detection in Spanish-language Social Networks using Sentiment Analysis Techniques" International Journal of Advanced Computer Science and Applications(IJACSA), 9(7), 2018. http://dx.doi.org/10.14569/IJACSA.2018.090733.

[14] Diego A. Andrade-Segarra and Gabriel A. Le´on-Paredes, "Deep Learning-based Natural Language Processing Methods Comparison for Presumptive Detection of Cyberbullying in Social Networks" International Journal of Advanced Computer Science and Applications(IJACSA), 12(5), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0120592