Identifying Human Aggressive Behaviour on Social Media Using Machine Learning Algorithms.

K M Bhavadharani,R Brindha Devi, S Ayishathul sameera
Student,
Department of Computer Science and Engineering,
Panimalar Institute of Technology,Poonamallae,chennai, Tamil Nadu,India
Prof.M.Geetha,B.E,
Panimalar Institute of Technology,Poonamallae,chennai, Tamil Nadu,India

Date of Submission: 12-03-2020 Date of Acceptance: 28-03-2020

I. INTRODUCTION

Cyberbullying is a sort of online badgering, which can be characterized as impolite, annoying, hostile, prodding, disheartening remarks through online internet based life focusing on one's instructive capabilities, sex, family and individual propensities. As per 'Tweens, Teens and Innovation 2014 Report' by McAfee[7], half of Indian Youth have had some involvement in cyberbullying. As per a review [10], it has been recognized that a noteworthy number of suicides have been submitted by youngsters who were presented to cyberbullying. Adolescents feel debilitated also, get baffled when they experience such cyber aggressive remarks which go about as an obstruction for investment also, mingling. Most systems administration locales today forbid the utilization of hostile and offending remarks. Be that as it may, this mostly being completed and separated to a constrained degree. As there is a gigantic measure of information accessible it is difficult to take the help of human arbitrators to physically hail each annoying furthermore, hostile remarks. Therefore, programmed classifiers that is quick and powerful to identify such sort of remarks is required which will additionally lessen cyberbullying. Notwithstanding, there are colossal difficulties required as remarks contains numerous exceptional characters eg: "You are a retard go post your head up your #%&*", "U r !diot" containing affronts and furthermore a few wry remarks. In this paper, we use AI methods to distinguish the affront and unpleasantness of the remarks present in informal communication destinations. The datasets utilized for tests are gathered from the Kaggle site.

The preparation datasets contain only 4000 remarks. The model is applied to the test set which contains near 2500 remarks. The primary goal is to foresee whether a remark is an affront to a member of a discussion. We have proposed two new theories for recognizing cyberbullying. Further, an examination between the exhibitions of mainstream AI characterization calculations is displayed. This issue is a paired characterization issue where we are attempting to characterize remarks as tormenting and non bullying. We have recognized highlights which distinguish hostile remarks guided towards peers notwithstanding standard highlights extraction methods, for example, TF-IDF score, N-grams, terrible word check and stemming to demonstrate Supervised AI calculations like Support vector machines, what's more, Logistic relapse. The element vector fabricated utilizing proposed includes adequately recognizes the remarks coordinated towards peers as tormented.

II. EXISTING SYSTEM

2.1 Data Collection

The website Formspring.me is a question and answer based website where users openly invite others to ask and answer questions. What makes this site especially prone to cyberbullying is the option for anonymity. Formspring.me allows users to post questions anonymously to any other user's page. To obtain this data, we crawled a subset of the Formspring.me site and extracted information from the sites of 18,554 users. The users we selected were chosen randomly. The number of questions per user in size ranged from 1 post to over 1000 posts. We also collected the profile information for each user. We were interested in first developing a language-based model for identifying cyberbullying, and the only fields we used in our study were the text of the question and the answer. Clearly a lot of rich information has been collected.

2.2 Labelling the Data

We extracted the question and answer text from the Formspring.me data for 10 files for the training set and 10 files for the testing set. These files were chosen randomly from the set of 18,554 users that were crawled, but we ensured that there was no overlap between the two sets of files. We used the same procedure to identify class labels both the training and the testing sets. We decided to use Amazon's Mechanical Turk service to determine the labels for our truth sets. Our training set contained 2696 posts; our test set contained 1219 posts. Our class labels were “yes” for a post containing cyberbullying and “no” for a post without cyberbullying. Of the 2696 posts in the training set, 196 received a final class label of “yes,” indicating the presence of cyberbullying. Of the 1219 posts in our test set, 173 were identified as cyberbullying, almost twice as many. These ratios confirmed our suspicion that the percentage of cyberbullying in the Formspring data was much higher than in other datasets that we’ve seen.

2.3 Developing the Model

We identified a list of insult and swear words, posted on the website www.noswearing.com. This list, containing 296 terms, was downloaded and each word on the list was given a severity level by our team. The levels were 100 (ex. idiot), 200 (ex. trash), 300 (ex. douchebag), 400 (ex. pussy), and 500 (ex. cuntass). The classification of these terms into severity levels was subjective. We were interested in both the number of “bad” words (NUM) and the density of “bad” words (NORM) as features for input to the learning tool. We therefore extracted two different training sets, one containing the count information, and one containing normalized information. We normalized by simply dividing the number of words at each severity level by the total number of words in the post, and then multiplying by 100 to get an integer value (for example, if there were 6 100-level words in a 10-word post, the 100-level would be reported as 60).

We also generated a feature to measure the overall “badness” of a post. We call this feature SUM and computed it by taking a weighted average of the “bad” words (weighting by the severity assigned). The SUM and TOTAL features were included in both the NUM and the NORM versions of our datasets. The class label (YES, NO) was also extracted from the Mechanical Turk file and included in the input to the machine learning tool.

2.4 Learning the Model

Weka is a software suite for machine learning that creates models using a wide variety of well-known algorithms. We identified the following algorithms as most useful for our project.

J48: The J48 option uses the C4.5 algorithm to create a decision tree model from the attributes provided. When working with decision trees, it is important to consider the size of the tree that is generated, as well as the accuracy of the model.

JRIP: JRIP is a rule-based algorithm that creates a broad rule set then repeatedly reduces the rule set until it has created the smallest rule set that retains the same success rate.

IBK: The instance-based (IBK) algorithm implemented in Weka is a k-nearest neighbor approach. We used the IBK method with k = 1 and k = 3.

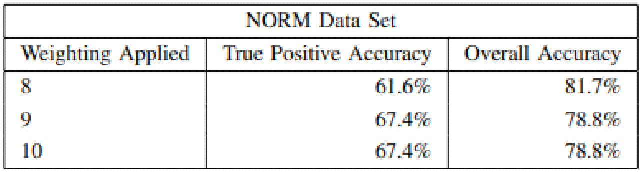
SMO: We wanted to use a support vector machine algorithm for testing also. Other teams that are working on similar projects found reasonable success with support vector machines. The SMO algorithm in Weka is a function-based support vector machine algorithm based on sequential minimal optimization.

2.5 Class Weighting

Less than 10% of the training data is positive (contained cyberbullying). As a result, the learning algorithms, by default, generated a lot of false negatives (i.e. they can reach accuracy figures of over 90% by almost ignoring the cyberbullying examples). We would prefer to have innocent posts labeled as cyberbullying (false positives) instead of mislabelling cyberbullying posts as innocent (false negatives). In order to overcome the problem of sparsity in the positive instances, we increased the weight of these instances in the dataset. We did this by simply copying the positive training examples multiple times in order to balance the training set and provide an incentive for the learners to identify the true positives.

2.6 Evaluation

We used two evaluation approaches in our experiments. We developed and labeled an independent test set using the same procedure that was used in the development of our training set. This set was used for testing of our most successful algorithm and the results appear below. However, the characteristics of the test set appear to be significantly different from the training set. More than twice as many posts were identified as cyberbullying. Additionally, both sets are relatively small and contain data from only 10 users of Formspring.me. For this reason, we also report statistics from experiments using cross-validation. Cross-Validation is an approach used to evaluate learning algorithms when the amount of labeled data available is small. Ten-fold cross-validation is considered to be the standard approach to evaluation in many machine learning experiments.



NORM Data Set		
Weighting Applied	True Positive Accuracy	Overall Accuracy
8	61.6%	81.7%
9	67.4%	78.8%
10	67.4%	78.8%

Figure 1:NORM Data Set Accuracy.

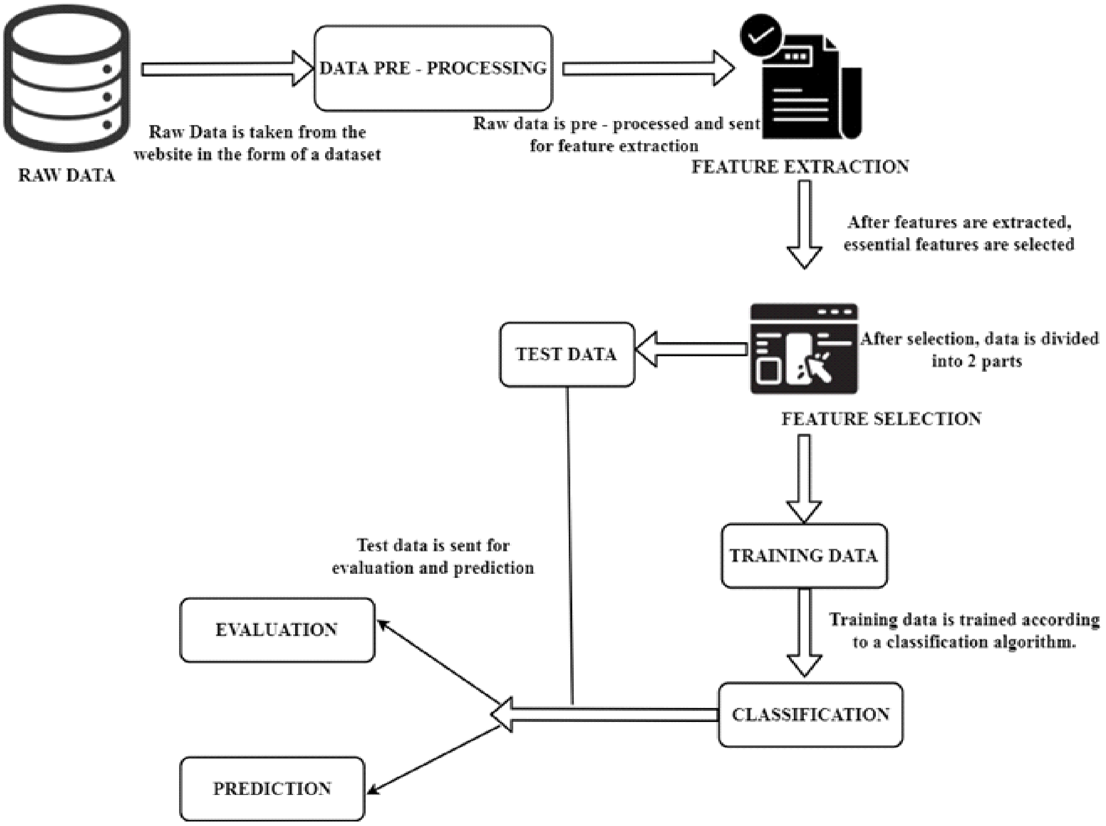
DISADVANTAGE

Although this language-based system correctly identifies 78.5% of the posts that contain cyberbullying in a small sample of Formspring data. The results indicate that our features do a reasonable job of identifying cyberbullying in Formspring posts. The considerable research effort is required to construct highly effective and accurate cyberbullying detection models and still, there is plenty of room for improvement on this timely and important application of machines learning to web data.

III. PROPOSED SYSTEM

The steps involved are Data Collection, Normalization, standard Feature extraction, additional feature extraction, feature selection and finally classification.

Architectural Diagram:


DATA PRE - PROCESSING
Raw Data is taken from the website in the form of a dataset
RAW DATA
Raw data is pre - processed and sent for feature extraction
FEATURE EXTRACTION
After features are extracted, essential features are selected
TEST DATA
After selection, data is divided into 2 parts
FEATURE SELECTION
Test data is sent for evaluation and prediction
TRAINING DATA
EVALUATION
Training data is trained according to a classification algorithm.
CLASSIFICATION
PREDICTION

3.1 Data Collection

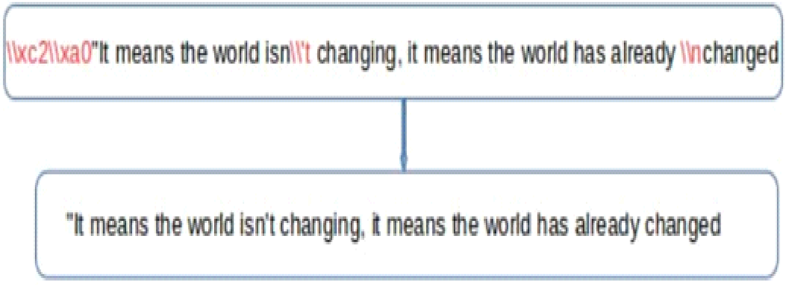
The datasets we use for our experiments are taken from the Kaggle website – an online competition site. The data consists of a label column followed by two attribute fields namely timestamps and Unicode escaped text of English language comment. The datasets contain training and test datasets.

3.2 Normalization

The Data set we have used contains a list of comments and respective labels. These should be converted into feature vector which is used by our machine-learning algorithms. For this, we use different Natural language processing techniques to obtain an accurate representation of the comments in the feature vector form. We use various techniques based on our observations.

3.2.1 Removing unwanted strings:

For the comments to be used by machine learning algorithms they should be in standard form. Raw comments present in a dataset that contains many unwanted strings like '\xa0','\\n' and many such encoding parts should be removed. Hence the first step is to pre-process the comments by removing unwanted strings, hyphens, and punctuations.

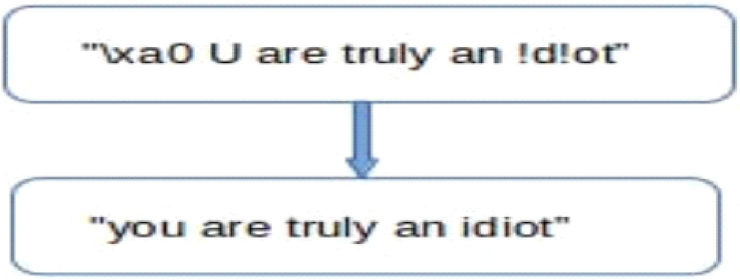


\\xc2\\xa0"It means the world isn\\'t changing, it means the world has already \\nchanged

"It means the world isn't changing, it means the world has already changed

Figure 2: Removal of unwanted strings.

3.2.2 Correcting Words: One of the reasons comments are classified as insulting is the presence of profane or abusive words. The total number of bad words present in comments is taken as one of the features. A dictionary of 500 bad words is compiled, which also includes variations of words (@$$, s h i t). This dictionary is used because people using the online forums sometimes use special characters to build an insulting word (!d!ot,@$$ole). When we encounter such words, the dictionary helps to convert them into natural forms. Also, Stemming is applied to capture bad word variations that are not contained in a dictionary. Stemming reduces a word to its core root, for example embarrassing is reduced to embarrass. Here it is noted that stemming is only applied to bad word dictionary, not on the dataset used, as it will lead to information loss. Again a small dictionary and a spell checker is used to convert all variations of “you”, “you’re” (e.g u, ur, etc) which are present in the dataset as participant use them as part of the flexible language.



"\xa0 U are truly an !d!ot"

"you are truly an idiot"

Figure 3: Correcting words that are in short forms.

3.3 Standard Feature Extraction

To train machine learning algorithms, strings should be converted in the feature vector. We use N-gram, counting and TF-IDF score to construct a feature vector. The process occurs in the following steps:

3.3.1N-gram model: N-grams are a group of a continuous sequences of n-items from a given text. These are used for dividing text and words into n chunks known as N-grams. Consider the sentence “You are funny” its unigram will be “you”, “are”, “funny”. Bigram- “you are”, “are funny”. Trigram- “funny you are”, “are funny you”. We use 2, 3, 4 and 5 N-grams for the building feature vector.

3.3.2 Counting: Count the number of times each of these tokens occurs in each of the text strings. This way we construct a sparse matrix of size N by V where N is the size of the training data which is number of comments and V is the size of the vocabulary, the length of feature vector constructed over the whole training set using n-grams, skip grams and use of pronouns representing all the text strings where the number of occurrences of each token is a feature for that text string.

3.3.3 TF-IDF Score: TF-IDF stands for “Term Frequency, Inverse Document Frequency”. It is a way to evaluate the importance of words (or “terms”) in a document based on how frequently they appear across various documents. The score signifies the importance of that term in relation to the original training data. TF-IDF score is given by: TF-IDF = tf_{ij} * idf_i.

Numerically, term frequency tf_{ij} specify the importance of a word i in comment j. It is determined as:

$$tf_{ij} = \frac{N_{ij}}{\Sigma N_j}$$

Where N_{ij} is the frequency of word i in comment j and ∑i is the frequency of all words in comment j. Inverse document frequency idf_i specifies the importance of a word i in the entire training dataset. It is determined as:

$$idf_i = \frac{\log |C|}{|C_j : W_i \epsilon C_j|}$$

Where |C| is the total number of comments, $|C_j{:}W_i^{\epsilon}C_j|$ is the number of comments where word W_i appears. So each comment contains a vector of words and each word is denoted in the vector by its TF-IDF score.

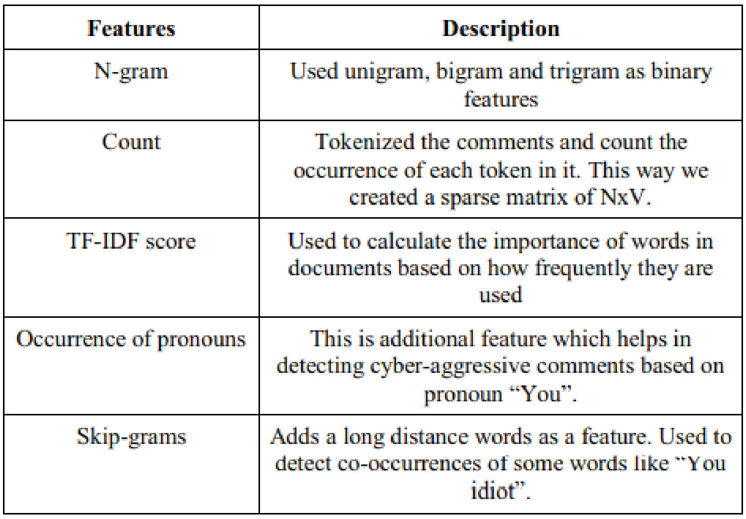
3.4 Additional Features

3.4.1 Capturing Pronouns:

It is been observed that cyber aggressive comments which are directed towards peers are perceived more negatively and results in cyberbullying. Comments containing a pronoun like 'you' followed by an insulting or profane word are peer-directed comments which are taken as negative and teens get frustrated after encountering such comments. So, to detect such comments we have used the count of pronouns as one of the features for detecting cyberbullying. To extract this feature we calculate the TF-IDF score of the pronoun present in the comment. This feature is our strong hypothesis which greatly increases the accuracy and helps in detecting cyber aggressive comments.

3.4.2 Skip – grams:

We also used skip-grams in building a feature vector as they help in detecting insult more effectively. These consider the long-distance words as a feature. For example consider “You are an idiot” as a comment, if we use 2 skip-gram, count of ‘You are’ as one feature and ‘an idiot’ as other is added in our feature – matrix. This way, the comments containing co-occurrences of words like “You idiots” which is negative and will be detected using skip – grams.



Features	Description
N-gram	Used unigram, bigram and trigram as binary features
Count	Tokenized the comments and count the occurrence of each token in it. This way we created a sparse matrix of NxV.
TF-IDF score	Used to calculate the importance of words in documents based on how frequently they are used
Occurrence of pronouns	This is additional feature which helps in detecting cyber-aggressive comments based on pronoun “You”.
Skip-grams	Adds a long distance words as a feature. Used to detect co-occurrences of some words like “You idiot”.

Figure 4: Features and their description.

3.5 Feature Selection:

The machine learning algorithms cannot handle all the features which are an order of some hundred thousand. So we need to select the best features out of our set of features. We use a statistical hypotheses method known as the “Chi-Squared test” to our feature matrix to select k best features where k is parameter roughly equal to 3000.

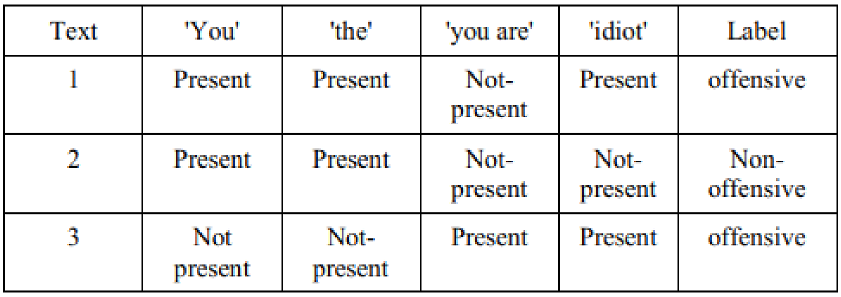
3.5.1 Chi – Square Method:

Chi-square (X2) method is commonly used for selecting the best features. This metric calculates the cost of a feature using the value of the chi-squared statistics with respect to class. Initially, a hypothesis H0 is assumed that the two features are unrelated, and it is The initial hypothesis H0 is the assumption that the two features are unrelated, and it is tested by chi-squared formula as is shown in equation:

$$X^2 = \frac{\Sigma(O_{ij} - E_{ij})}{E_{ij}}$$

Where O_{ij} is the observed frequency and E_{ij} is the expected frequency, asserted by the null hypothesis. Higher the value of (X^2), greater the evidence against the hypothesis H_0, hence more related is the two variables. Lesser the value of (X^2), the hypotheses tends to be true, the variables are independent.

To understand this measure better, consider the following example:



Text	'You'	'the'	'you are'	'idiot'	Label
1	Present	Present	Not-present	Present	offensive
2	Present	Present	Not-present	Not-present	Non-offensive
3	Not present	Not-present	Present	Present	offensive

Figure 5: Example

Considering ‘you’ and ‘idiot’ be both independent, then expected number of rows where these happen to be present is given by:

$$E('you\ present', offensive) = \frac{N('you\ present') * N('offensive')}{N}$$

Where N (‘you present’, offensive) is the number of rows which have the feature ‘you’ and are labelled as offensive and N is total number of rows.

$$X^2 = \frac{\Sigma(observed\ (i,j) - Expected(i,j))^2}{Expected(i,j))}$$

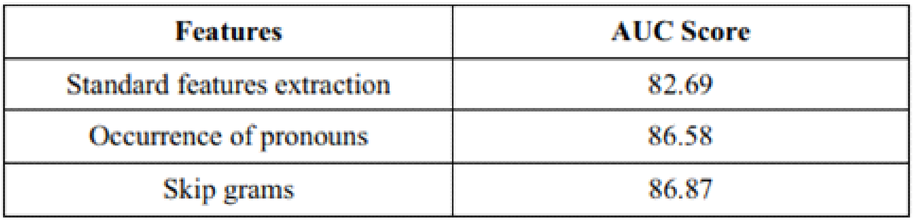
Where i = {‘yes present’, ‘yes not present’} and j = {‘offensive’, ‘non offensive’}. Higher these values more related are two variables.

3.6 Classification

Once the features are built, we extract the best features using the chi-squared test and apply the machine learning algorithms to train models on it. We have used SVM and logistic regression on our feature data.

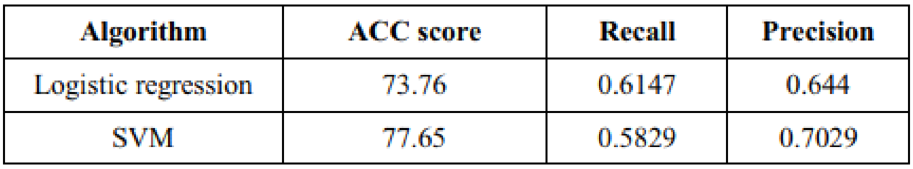
We get the final results by combining the results obtained by both algorithms. The final output is the probability of comment being insulting. The test dataset which is classified contains 2647 comments.

At first, we build a feature vector containing standard feature extraction containing TF-IDF and N-grams. Then we train our algorithms based on these feature vector and the best accuracy achieved is of logistic regression of 83%. Then, we include the occurrence of pronouns and skip-gram as features which increased the accuracy and logistic regression outperformed in this too with 86%. The test datasets used for our experiment contained nearly 3000 unlabeled comments. Also, we tried to train the system with all features using SVM and logistic regression. An experimental result suggests that comments targeted towards peers help in detecting cyberbullying more efficiently. The table shows the performance of algorithms trained on the standard features extraction techniques. The table shows the accuracy (AUC score), precision and recall values after introducing skip grams and pronouns as features. The following figure shows the increase in accuracy by introducing additional features in addition to traditional features.



Features	AUC Score
Standard features extraction	82.69
Occurrence of pronouns	86.58
Skip grams	86.87

Figure 6



Algorithm	ACC score	Recall	Precision
Logistic regression	73.76	0.6147	0.644
SVM	77.65	0.5829	0.7029

Figure 7

Skips	*AUC score*	*Recall*	*Precision*
2 skips	86.84	0.72	0.765
3 skips	86.84	0.71	64.64
2,3 skips	86.92	0.71	0.769

Figure 8

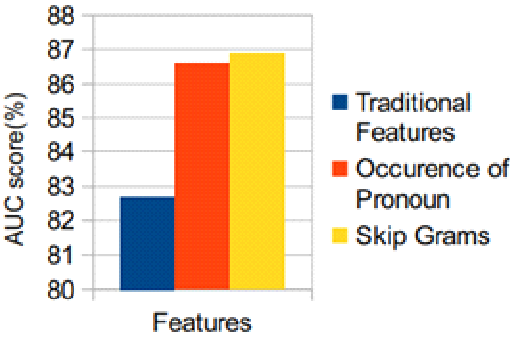


Figure 9 :AUC Score

3.7 ADVANTAGE

Two new hypotheses for feature extraction are developed which helps detect cyberbullying. A model is built which predicted comments as bully/nonbully. The result is the probability of comment being offensive to participants. The hypothesis increases the accuracy by 4% and can be used to detect the comments that are targeted towards peers.

IV. CONCLUSION

In this paper, we presented two new hypotheses for feature extraction which can be helpful in detecting cyberbullying. We built a model that predicted comments as bully/nonbully. The end result is the probability of comment being offensive to participants. Results show that our hypothesis increases the accuracy by 4% and can be used to detect the comments that are targeted towards peers. Future work should be directed towards detecting sarcastic comments.

REFERENCES

[1]. Al-garadi Mohammed Ali, Hussain, M. R., Khan, N., Murtaza, G., Nweke, H. F., Ali, I., … Gani, A. (2019). Predicting Cyberbullying on Social Media in the Big Data Era Using Machine Learning Algorithms: Review of Literature and Open Challenges. IEEE Access, 1–1.

[2]. Reynolds, K.; Kontostathis, A.; Edwards, L., "Using Machine Learning to Detect Cyberbullying," Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on, vol.2, no.,pp.241,244,18-21Dec.2011.

[3]. K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the Detection of Textual Cyberbullying," in Proc. IEEE International Fifth International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 2011.

[4]. Spertus, E., Smokey: Automatic recognition of hostile messages. In: Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence, pp. 1058–1065 (1997).

[5]. Mahmud, A., Ahmed, K.Z., Khan, M, Detecting flames and insults in text. In: Proceedings of the Sixth International Conference on Natural Language Processing (2008)

[6]. M.Fire,R.Goldschmidt,andY. Elovici,“Online SocialNetworks:Threats andSolutions,”IEEECommun.SurveysTut.,vol.16,no.4,pp.2019–2036, Oct.–Dec.2014.

[7]. B. Alghamdi, J. Watson, and Y. Xu, “Toward Detecting Malicious Linksin Online Social Networks Through User Behavior,” in Proc.IEEE/WIC/ACM Int. Conf. Web Intell. Workshops, 2016, pp.5–8.

[8]. X.Ruan,Z.Wu,H.Wang,andS.Jajodia,“ProfilingOnlineSocialBehaviors forCompromisedAccountDetection,”IEEETrans.Inf.ForensicsSecurity, vol. 11, no. 1, pp. 176–187, Jan.2016.

[9]. Z. Yang, J. Xue, X. Yang, X. Wang, and Y.Dai,“Votetrust:Leveraging Friend Invitation Graph to Defend Against Social Network Sybils,” IEEE Trans. Depend. Secure Comput., vol. 13, no. 4, pp. 488–501, Jul./Aug.2016.

[10]. U.U.S.Khan,M.Ali,A.Abbas,S.Khan,andA.Zomaya,“Segregating SpammersandUnsolicitedBloggers fromGenuineExpertsonTwitter,”IEEE Trans. Depend. Sec. Comput.,2016.

[11]. N.Laleh,B.Carminati,andE.Ferrari,“RiskAssessmentinSocialNetworks basedonUserAnomalousBehaviour,”IEEETrans.Depend.Sec.Comput., 2016.

[12]. J. Zhang, R. Zhang, Y. Zhang, and G. Yan, “The Rise of Social Botnets: AttacksandCountermeasures,”IEEETrans.Depend.Sec.Comput.,2016.

[13]. S. Cresci, R. D. Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "SocialFingerprinting:DetectionofSpambotGroupsThroughDNA-inspired Behavioral Modeling," IEEE Trans. Depend. Secure Comput.,2017.

[14]. G.Yang,S.He,andZ.Shi,"LeveragingCrowdsourcingforEfficientMaliciousUsersDetectioninLarge-ScaleSocialNetworks,"IEEEInternetThings J., vol. 4, no. 2, pp. 330–339, Apr.2017

[15]. N. Z. Gong, M. Frank, and P. Mittal, "Sybilbelief: A Semi-Supervised Learning Approach for Structure-Based Sybil Detection," IEEE Trans. Inf. Forensics Security, vol. 9, no. 6, pp. 976–987, Jun.2014.

K M Bhavadharani "Identifying Human Aggressive Behaviour on Social Media Using Machine Learning Algorithms." *International Journal of Computational Engineering Research (IJCER*), vol. 10, no.1, 2020, pp 01-08.