Hate Speech Detection Using Modified Principal Component Analysis and Enhanced Convolution Neural Network on Twitter Dataset

Majed Alowaidi

Abstract Traditionally used for networking of computers and communications, the Internet has been evolving from the beginning. Internet is the backbone for many things on the web including social media. The concept of social networking which stood in the early 1990s has also been growing with the internet. SNSs (Social Networking Sites) sprung and stayed back to an important element of internet usage mainly due to the services or provisions they allow on the web. Most people use SNSs like Twitter or Facebook as their own medium of expressions and speech. These sites allow posting of photos, videos and support audio and video storages on the sites which can be shared amongst users. Though an attractive option, these provisions have also culminated in issues for SNSs like posting offensive material. Though not always, users of SNSs have their share in promoting hate by their words or speeches which is difficult to be curtailed after being uploaded in the media. Hence, this work proposes to identify hate speeches in user’s reviews from the twitter dataset. The work uses MPCA (Modified Principal Component Analysis) and ECNN (Enhanced Convolution Neural Network) for identifying hate speeches. NLP (Natural Language Processing) is implemented to build an automatic system for syntactic and semantic analysis. This proposed work contains main phases such as pre-processing, feature extraction and classification process. The pre-processing is done by using normalization method which is used to remove the white spaces, replace the consecutive exclamation and question marks, and eliminate stop words. These preprocessed features are taken into feature extraction process. MPCA algorithm is applied to perform feature extraction process. It uses set of correlated features and extracts more informative features for the given dataset. Then the classification algorithm is proposed to detect the hate speech or abusive languages. ECNN is proposed to classify hate
and non-hate from the online content more accurately. It takes many inputs and generates output with minimum amount of time with higher accuracy for larger dataset. Thus, the result concludes that the proposed MPCA+ECNN algorithm provides higher accuracy, precision, recall and F-measure values rather than the existing methods.

**Keywords** Hate speech detection · Natural Language Processing (NLP) · Modified Principal Component · Modified Principal Component Analysis (MPCA) · Enhanced Convolution Neural Network (ECNN)

1 INTRODUCTION

SNSs are an attractive medium for sharing important information. The proliferation of the internet by SNSs and parallel spurt in its usage has facilitated societal and familial growths. This medium has also thrown open opportunities for people around the globe to express themselves without fear. This has also resulted in the circulation of threats and abusive language called Cyber bullying, a form of bullying using the medium of Internet. Hate speech is kind of words or sentences used by people who opine without any fear, but a little vulgarly. Hate speech can be targeted against someone where other are also led towards hating the target[1][2]. SNSs have become an easy medium for propagating and breeding hateful content which can ultimately lead to heinous crimes. Moreover, the capability to maintain anonymity in such posts or speeches has made it easy for anonymous aggressive communications. Fig 1 depicts hate speech reviews in social media.

![Fig. 1 Hate speech review in social media](image)
Currently, online hate speeches are growing day by day making its automatic detection compulsory. Studies on SNS security have also been growing in substantial numbers [3] as these sites have become a great source for analysis due to the voluminous data they generate. The sites are also being exploited for undesired activities or in this case hateful speeches [4]. People who indulge in these kinds of activities manage to get away easily as most web operations are anonymous and hide the identity of its origin. SNSs also do not concentrate on removing these kinds of speeches or words. One alternative however is automation of tasks which analyze, recognize and eliminate vulgarity in words or audio. Thus, the uses of NLPs and MLTs (Machine Learning Techniques) have grabbed the attention of academicians and researches [5][6]. In spite of improvements in the field, issues arise in variability of data and datasets and reduced evaluation competitiveness [7].

Studies proposing Feature extractions using MLTs s cater to some kind of derivations from input features for identifying distinctive, non-repetitive and informative properties in features [8]. These operations result in generating a subset of features of relevance from the original feature list. Though it is complex to arrive at such subsets, predictive models built on such subset of features have shown remarkable success in terms of accuracy or classifications. Typical methodologies used in SMAs (Social Media Analytics) are N-grams, Bag of words, TFIDF (Term Frequency-Inverse Document Frequency), semantic/syntactic analytics, dictionaries and parts of speech.

Classification or categorization of Tweets can be based on pre-existing classes where prior steps include pre-processing, feature extractions/decomposition [9]. These sub-processes are carried out in proposed schemes for improvising classification accuracies. MLTs classify data based on a training/learning process. SVM (Support Vector Machine) is a supervised MLT used in classifications of data. SVM classifications result into words being categorized as hate speech or normal words when applied to the hate speech dataset. SVMs are simple binary classifiers which constructs a hyper plane by separating class members from the input space. SVMs also use a non-linear mapping function which maps input space values to a feature based high dimensional space. The planes are separated by a maximum margin hyper plane which is a linear combination of data points [10]. In the process, SVMs also identify informative points (Support Vectors) to represent its separating hyper plane. Figure 2 shows the model of hate speech detection.

The main issue undertaken for this research work is the problem of detecting hate speeches in twitter using a dataset. This work uses MPCA and ECNN to improve the hate speech detections from datasets. The next section is a review of literature while the proposed schemes are detailed in section 3 followed by results and analysis and conclusion.
2 RELATED WORK

Hate speeches against minorities was detected by the approach proposed in [11]. The study collected and processed Facebook data following regular processing steps. They used Word2Vec a word embedding technique and n-grams. Their identified features were then classified using DNNs (Deep Learning Techniques) like GRU (Gated Recurrent Unit) and variants of RNNs (Recurrent Neural Networks). The identified hate words were clustered using Word2Vec to predict targets of hatred. They experimented their techniques in Ethiopian, Amharic texts. A customized dataset was created by crawling Face book pages as the texts could not be found in regular trainable data sets. Their feature extractions using Word2Vec performed better than other classical methods in experimentations while their DLTs provided better classification accuracy.

Crowd sourcing was used in [12] where the authors used it to collect hate speech tweets. The crowd-source lexicons categorized tweets into offensive, hate and normal language words. A multi-class classifier distinguished these categories. Their minute analysis differentiated offensive language from hate speeches effectively. They found racist and homophobic tweets have hate speech content while sexist tweets are offensive.

PCA (Principal Component Analysis) was optimized in [13] for extracting features. The technique made use of parallel co-ordinates of multivariate graphical plots. The techniques achieved twin objectives of automation and
filtering processes which were executed manually. Variables with bigger variances when extracted do not help multi-variate classifications, but the study’s use of PCA in feature extractions overcame the issue. Their proposal’s use of parallel co-ordinate plots achieved better performance while being tested with vegetable oil data. The study also implies this technique could be applied most feature extraction methods which needed to classify multi-variate data.

The study in [14] applied LDA (Linear Discriminant Analysis) to extract useful information from features discarded by PCA which discards features with marginal variance in class constructions. These features may carry useful information and hence their features are extracted by LDA. They called the scheme PDCA (Principal Component Discriminant Analysis) improved te accuracy of classifiers when compared to PCA or LDA. The study’s experiments on an urban and agriculture image showed its efficiency.

A supervised approach for extracting n-grams from tweets was proposed in [15]. The study reduced features using n-grams and statistical analysis with which a Twitter-specific lexicon for SA (Sentiment Analysis) was developed. The lexicon considered only brand related terms in tweets thus reducing modeling complexity but maintaining wide coverage of topics. To demonstrate their method’s usefulness the reduced lexicon was compared with a traditional sentiment lexicon in classifications using SVM. Their results showed significant improvements in recall and accuracy metrics. DAN2 machine learning also proved the reduced lexicon’s utility in text classifications by producing more accurate sentiment classification results than SVM.

CNN (Convolution Neural Networks) figured in the study of [16] which investigated pre-trained CNNs efficacy to search large environmental datasets. Their investigations showed DLTs could perform better in image recognitions and classifications. Training these neural networks over large imagery datasets facilitated many applications like content-based image searching and retrieval as the fidelity provided by the convolution filters of CNNs could promote accurate content searches within voluminous datasets.

**PROPOSED METHODOLOGY**

Though studies have detected hate speeches their accuracy of detections can be enhanced. Moreover, elongated execution times and reduced accuracies in classifications have been the main motivational factor for this research work. This work proposes MPCA+ECNN to improve overall performance of hate speeches. This proposed work used uses MPCA algorithm for extracting important features which are then classified by the proposed ECNN.

Proposed **Pre-processing**

The dataset values are pre-processed using normalization as noisy or unclean data impact overall classification accuracies while reducing feature vector spaces also which reduces execution times of the process. The steps followed in pre-processing are Word segmentation, Stemming and stop words removal. Tweets sentences are broken into words called segmentations. This works uses NLTK (Natural Language Toolkit) tokenizer to split tweets into words. Different words can be viewed in a singular form or verbs and nouns can be mapped as a semantically similar word. Stemming is removing prefixes and
suffixes of a word resulting in a stemmed word. They may be grammatically wrong but work wonders in classifications. Some methods use lemmatization additionally, but time complexity gets increased [17]. This work uses a Porter Stemmer for stemming tweet words.

Tweets have many unimportant words which get repeated often and do not add value for classifications and thus need to be eliminated. This work uses NLTK library [18] for removing stop words and to reduce the impact of stop words in twitter sentiment classification. Normalization is the final step of pre-processing in this work. This work uses Min-Mix Normalization, a technique which linearly transforms data [19] and fits it into a pre-defined boundary. This normalization can be depicted as Equation (1)

\[
A' = \left( \frac{A - \text{minvalue of } A}{\text{maxvalue of } A - \text{minvalue of } A} \right) \times (D - C) + C
\]  

(1)

Where, \(A'\) - Normalized data output, \([C,D]\) Aboundary and \(A\) – data to be transformed. The transformations done in this method use mean and SD (Standard Deviation) for obtaining normalized values. Normalization followed in this work aims at removing non-informative and noisy features.

2.1 Feature extraction using MPCA algorithm

Feature extraction processes are used to retrieve informative features from data or datasets. PCA is a multivariate data analysis method that is used in linear feature extractions. PCA identifies correlations between features in data called observed variables. It ignores features with minor variations generating a feature subset. Thus, it reduces dimensionality in a feature space of observed variables and extracts correlated variables to form a reduced feature space [20] [21]. In this work, PCA is used to generate feature vectors from the hate speech dataset. One issue in using PCA is its ignoring features with minor variations of correlations which may carry important feature information. This issue is overcome in this work by modifying the PCA algorithm with the proposed MPCA approach. MPCA reduced eigenvector value influences by normalizing the vectors. Assuming \(y_{ij}\) is the \(j^{th}\) element of the \(i^{th}\) feature vector, then SD \(\sqrt{\lambda_j}\) can be applied on the feature vector. The resulting feature vector \(y'_i\) can be rewritten as Equation (2)

\[
y'_i = \begin{bmatrix} y_{i0} \sqrt{\lambda_0}, & y_{i1} \sqrt{\lambda_1}, & \ldots, & y_{ir-1} \sqrt{\lambda_{r-1}} \end{bmatrix}
\]  

(2)

These resulting normalization on feature vectors, create a new feature subspace. This study normalizes feature vector values by their corresponding eigenvalue square roots followed by calculations of training and testing feature distances. This linear transformation of PCA is depicted in Equation (3)

\[
Y = TX
\]  

(3)
Where, \( T \) - transform matrix, \( X \) - feature vectors and \( Y \) - Transformed feature vectors. Transform matrix \( T \), uses Equation (4)

\[
(\lambda I - S) U = 0 \tag{4}
\]

Where the square matrices: \( I \) - has identity values in its diagonal; \( S \) - covariance matrix of original data, \( U \) and \( \lambda - \text{eigenvectors} \). Eigen values \( U_j \) and \( \lambda_j \) \((j = 1, 2, \ldots, m)\) is computed using (2) in the order \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \). The eigenvectors \( U \) can be expressed as \( U = [U_1, U_2, \ldots, U_m] \).

In the proposed MPCA transformed the Twitter dataset into \( Y \), the transformed matrix in the training samples and expressed in Equations (5) (6) and \((7)

\[
Y = T^T X \tag{5}
\]

\[
V_N = b_1 u_1 + b_2 u_2 + \ldots + b_N u_N \tag{6}
\]

\[
S = \sum_{i=1}^{1} b_1 u_1; 1 < N \tag{7}
\]

Comparing (3) and (5), transformed matrices arise from the covariance matrix and complete hate speech dataset.

MPCA’s main advantage is dimensionality reduction and reduction of information loss. Though based on PCA, it is a mathematical procedure that maps high dimensional data to lower dimensional data using a linear transformation where lower dimensions are defined by Eigen vectors of the covariance matrix. Thus, MPCA extracts hate speech features with reduced errors.

**Algorithm 1: MPCA**

1. Start
2. Find the mean value \( S' \) of the given Twitter dataset \( S \)
3. Subtract the mean value from \( S \)
4. Obtain the new matrix \( A \)
5. Covariance is obtained from the matrix i.e., \( C = AA^T \) Eigen values are obtained from the covariance matrixes that are \( V_1 V_2 V_3 \ldots V_N \)
6. Finally, Eigen vectors are calculated for covariance matrix \( C \)
7. Any vector \( S \) can be written as linear combination of Eigen vectors using (6)
8. Only Largest eigen values are kept to form lower dimension data set
9. Match the combination of words in the given tweets (7)
10. Extract the more informative features (tweets)
11. End
2.1.1 ECNN algorithm for hate speech detection

In this proposed work, Improved Convolutional Neural Network (ECNN) is introduced to classify the test data into yes (or) no classes. The proposed deep learning method attains higher accuracy. The basic CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers and fully connected layers. Convolutional layers apply a convolution operation to the input, transferring the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli. The architecture diagram of ECNN is shown in Fig 3.

![Architecture diagram of ECNN](image)

CNNs may incorporate local or global pooling layers which join neuron’s cluster outputs into the neuron of the subsequent layer. Mean pooling is implementing neuron cluster average values in the previous layer while Fully connected layers map each neurons of layers to neurons in other layers. CNNs are principled like traditional MLP (Multi-Layer Perceptron) neural networks [22]. This work’s ECNN has input/convolution/ sub-sampling/ classification layers and can analyze high-dimensional data efficiently. Parameters are shared in convolution layers reduce the number of parameters.

ECNN’s input layer gets its input (Tweets) from samples which are then transformed and submitted to the next layer. Initial parameters like local fields scale and filters are also defined in this layer.

Cx (Convolution layer) convolutes inputs for producing several layers called feature maps which include previous convolution layer computations. This layer extracts key features and reduces the computational complexity of the network.

An activation function is executed in the convolution layer. The function maps outputs to a set of inputs creating a non-linear network structure. Weights are added to feature values for a new pattern output defined as...
Equations (8) and (9)

\[ y(n) = f(\sum_{i=1}^{N} w_i(n) x_i(n)) \]

(8)

where, \( f(x) = \begin{cases} 
+1 & \text{if } x \geq 0 \\
-1 & \text{if } x < 0 
\end{cases} \) (9)

where \( n \) is the iteration index

Connection weights are updated according to

\[ w_i(n + 1) = w_i(n) + \eta(d(n) - y(n)) x_i(n), \quad i = 1, 2, \ldots N \] (10)

Where \( \eta \) is the gain factor

Then apply standard deviation

\[ \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{N} f_i(x_i - \overline{x})^2} \] (11)

The weighted features are fed to ECNN for better classification accuracy. A Polynomial distribution function ensures that the same set of data is analyzed. Every feature map generated in a convolution layer is sub-sampled in this layer. ECNN, classifies tweets as hateful or normal by using a genetic fitness value for selections as they identify accurate features based on the best fitness value. The proposed CNN is enhanced by polynomial distribution and application of genetic fitness values. Two sample are selected for the genetic operator’s parents which then evolve next generation chromosomes. This is repeated until the best fitness scores are found. The selection operator for random selections can be defined as Equation (12)

\[ P(c_i) = \frac{f(c_i)}{\sum_{j=1}^{n} f(c_j)} \] (12)

Where, \( P(c_i) \) - probability of a chromosome \( c_i \) with population size \( n \). This implies chromosomes with high fitness scores are selected which also implies children have more fitness after a crossover operation.

**Algorithm 2: Steps in ECNN**

Input: Twitter Dataset, population size, training data, testing data

Objective function: Assign the accuracy as fitness function

Output: Hate or non hateful tweets

1. Start

2. Procedure Twitter dataset

3. For all input attribute, describe hate speech attribute \( \in \) Twitter dataset

   do

4. Initialize the population

5. While \( t < \text{max generation number} \) do

6. Convert the input into sub layers

7. Detect hate or non hateful tweets features using (10) and (11)
8. Compute fitness function values using (12)
9. Select more informative and relevant features
10. Perform training and testing process for given dataset
11. \( t = t + 1 \)
12. Return the highest detection accuracy features
13. Copy predefined hate speech feature label for each feature as per the input dataset
14. End

While evaluating the fitness of each individual in ECNN, after fit parents are selected, they generate new offspring due to genetic operators while increasing the counter by one and once maximal generations are reached, the operations stop. Overall work flow of the proposed system is depicted in Figure 4.
Fig. 4 Overall block diagram of the proposed system
3 Experimental Result and Discussion

This section displays the results of the proposed technique. Publicly available
hate speech tweets dataset, compiled and labeled [23] was used in the study.
The dataset has 14509 tweets samples divided into three distinct classes
(offensive, non-offensive and hate speech) number of tweets where sixteen
percent are hate speeches, thrity three percent are offensive without hate
speech while the balance are normal non offensive tweets. The proposed work
was compared with SVM and RNN classification of hate speeches in terms of
precision, recall and f-measure.

**Performance Measures Used:**

**Precision:** Precision is finding how precise/accurate a model is from
predicted positive to how many of them are actually positive. Precision is
a good measure to determine, when the costs of False Positive is high.

\[
\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}
\]

**Recall:** Recall calculates how many of the Actual Positives a model capture
through labeling it as Positive (True Positive). Recall shall be the model
metric to select the best model when there is a high cost associated with
False Negative.

\[
\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}
\]

\[
= \frac{\text{True Positive}}{\text{Total Actual Positive}}
\]

**F1 Score:** F-measure is a measure of a test’s accuracy

\[
F1 - \text{score} = \frac{1}{2 \left( \frac{1}{\text{recall}} + \frac{1}{\text{precision}} \right)}
\]

**Accuracy:** It is the percentage of the test tuples that are classified properly
by any algorithm.

\[
\text{Accuracy} = \frac{\# \text{ of true positives} + \# \text{ of true negatives}}{\# \text{of true positives} + \text{false positives} + \text{true negatives}}
\]

The comparative performance of classification methods with respect to precision
is depicted in Figure 5.

It is evident from the above figure that SVM or RNN algorithms have
lower precision values when compared with MPCA+ECNN implying that
the proposed method identifies hate speech better with higher precision.
Comparative performance of classification methods with respect to recall is
depicted in Figure 6.
Fig. 5 Precision

Fig. 6 Recall
It is evident from the above figure that SVM or RNN algorithms have lower recall values when compared with MPCA+ECNN implying that the proposed method identifies hate speech better with higher recall values. Comparative performance of classification methods with respect to F-measure is depicted in Figure 7.

![Fig. 7 F-measure](image)

It is evident from the above figure that SVM or RNN algorithms have lower F-measure values when compared with MPCA+ECNN implying that the proposed method identifies hate speech better with higher recall values. Comparative performance of classification methods with respect to Accuracy is depicted in Figure 8.

It is evident from the above figure that SVM or RNN algorithms have lower F-measure values when compared with MPCA+ECNN implying that the proposed method identifies hate speech better with higher recall values. Thus, it can be inferred that the proposed MPCA+ECNN out beats the other algorithms with its superior performance in detecting hate speech from hate speech dataset of Twitter.

**CONCLUSION**

To summarize, this paper proposes identifying hate speech in users’ words in SNSs space. This is achieved by pre-processing the hate speech dataset by changing tweet text into lower case and cleans the dataset by eliminating URLs, white spaces, usernames, hash tags, stop-words and punctuations. The tweets are then tokenized for stemming. Pre-processing in this work ends with a normalization technique used on the dataset samples. Important features are extracted using a modified PCA which is then passed to ECNN for classification. The performance of the proposed MPCA+ECNN demonstrates
higher performances in the areas of precision, recall, accuracy and F-measure when benchmarked with SVM and RNN. It can be concluded that the proposed work is viable for implementations of detecting hate speeches in Twitter. Future scope would be to propose optimizations for handling voluminous of datasets using fuzzy clustering. Also, this work can be extended to include more SNSs networks.

Funding No funding source available.

Conflicts of interest/Competing interests None

Availability of data and material https://data.world/crowdflower/hate-speech-identification

Code availability N/A

REFERENCES

1. Hajime Watanabe, Mondher Bouazizi, and Tomoaki Ohtsuki. Hate speech on twitter a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection.IEEEAccess, 2 2018. ISSN 2169-3536.
2. Ziqi Zhang, David Robinson, and Jonathan Tepper. Detecting hate speech on twitter using a convolution gru based deep neural network. In15th European Semantic Web Conference 2018, pages 745–760. Springer, 2018.
3. Shervin Malmasi and Marcos Zampieri. Detecting hate speech in social media. 2017.
4. Biere, Shanita, Sandjai Bhulai, and Master Business Analytics. "Hate speech detection using natural language processing techniques," Master Business AnalyticsDepartment of Mathematics Faculty of Science (2018).
5. Paula Fortuna. Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes. 2017.
6. Robinson, David, Ziqi Zhang, and Jonathan Tepper. "Hate speech detection
on Twitter: feature engineering vs feature selection." European Semantic Web Conference. Springer, Cham, 2018.

Gomez, Raul, et al. "Exploring hate speech detection in multimodal publications." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020.

8. Amit G. Shirbhate, Sachin N. Deshmukh. "Feature Extraction for Sentiment Classification on Twitter Data," International Journal of Science and Research, ISSN: 2319-7064, Volume 5 Issue 2, February 2016

9. Batool, Rabia, et al. "Precise tweet classification and sentiment analysis." 2013 IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS). IEEE, 2013.

10. Ahmad, Munir, Shahib Aftab, and Iftikhar Ali. "Sentiment analysis of tweets using svm." Int. J. Comput. Appl 177.5 (2017): 25-29.

11. Mossie, Zewdie, and Jenq-Haur Wang. "Vulnerable community identification using hate speech detection on social media." Information Processing & Management 57.3 (2020): 102087.

12. Davidson, Thomas, et al. "Automated hate speech detection and the problem of offensive language." Proceedings of the International AAAI Conference on Web and Social Media. Vol. 11. No. 1. 2017.

13. Haibo, Gao, et al. "Optimization of principal component analysis in feature extraction." 2007 International Conference on Mechatronics and Automation. IEEE, 2007.

14. Imani, Maryam, and Hassan Ghassemian. "Principal component discriminant analysis for feature extraction and classification of hyperspectral images." 2014 Iranian Conference on Intelligent Systems (ICIS). IEEE, 2014.

15. Ghiassi, Manoochehr, James Skinner, and David Zimbra. "Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network." Expert Systems with applications 40.16 (2013): 6266-6282.

16. Freeman, J. "Content search within large environmental datasets using a convolution neural network." Computers & Geosciences 139 (2020): 104479.

17. Riordan, Brian, and Michael N. Jones. "Redundancy in perceptual and linguistic experience: Comparing feature-based and distributional models of semantic representation." Topics in Cognitive Science 3.2 (2011): 303-345.

18. Farkiya, Alabhya, et al. "Natural Language Processing using NLTK and WordNet." (2015): 5465-5469.

19. Jain, Sukirty, Sanyam Shukla, and Rajesh Wadhvani. "Dynamic selection of normalization techniques using data complexity measures." Expert Systems with Applications 106 (2018): 252-262.

20. Taguchi, Y. H., and Yoshiki Murakami. "Principal component analysis based feature extraction approach to identify circulating microRNA biomarkers." PloS one 8.6 (2013): e66714.

21. Ankita Pal, "Principal Component Analysis of TF-IDF In Click Through Rate Prediction," International Journal of New Technology and Research (IJNTR).ISSN: 2454-4116, Volume-4, Issue-12, December 2018, pp 24-26

22. Suganuma, Masanori, Shinichi Shirakawa, and Tomoharu Nagao. "A ge-
netic programming approach to designing convolutional neural network architectures." Proceedings of the genetic and evolutionary computation conference. 2017.
23. https://data.world/crowdflower/hate-speech-identification