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

Majed Alowaidi †
m.alowaidi@mu.edu.sa
Department of Information Technology, College of Computer and Information Sciences, Majmaah University, 11952, Majmaah, Saudi Arabia

Abstract
Traditionally used for networking 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 started in the early 1990s has also been growing with the internet. Social Networking Sites (SNSs) sprung and stayed back to an important element of internet usage mainly due to the services or provisions they allow on the web. Twitter and Facebook have become the primary means by which most individuals keep in touch with others and carry on substantive conversations. These sites allow the posting of photos, videos and support audio and video storage on the sites which can be shared amongst users. Although an attractive option, these provisions have also culminated in issues for these sites 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 article outlines a process for extracting user reviews from the Twitter corpus in order to identify instances of hate speech. Through the use of MPCA (Modified Principal Component Analysis) and ECNN, we are able to identify instances of hate speech in the text (Enhanced Convolutional Neural Network). With the use of NLP, a fully autonomous system for assessing syntax and meaning can be established (NLP). There is a strong emphasis on pre-processing, feature extraction, and classification. Cleansing the text by removing extra spaces, punctuation, and stop words is what normalization is all about. In the process of extracting features, these features that have already been processed are used. During the feature extraction process, the MPCA algorithm is used. It takes a set of related features and pulls out the ones that tell us the most about the dataset we give itThe proposed categorization method is then put forth as a means of detecting instances of hate speech or abusive language. It is argued that ECNN is superior to other methods for identifying hateful content online. It can take in massive amounts of data and quickly return accurate results, especially for larger datasets. As a result, the proposed MPCA+ECNN algorithm improves not only the F-measure values, but also the accuracy, precision, and recall.

Keywords:
Natural Language Processing, Modified Principal Component An alysis, Hate speech detection and Enhanced Convolution Neural Network.

Manuscript received January 5, 2023
Manuscript revised January 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.1.15

1. Introduction

Social networking sights have been popular due to growing age of various services applications in education, business and marketing. Therefore, these websites like Facebook, Twitters, and LinkedIn etc. have been provide useful and attractive medium for sharing important information for growing the business. 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 Cyberbullying, a form of bullying using the medium of the internet. Hate speech is/are words or sentences used by people who opine without any fear, but a little vulgarly. People can be the target of hate speeches in which the audience is encouraged to hate the speaker [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

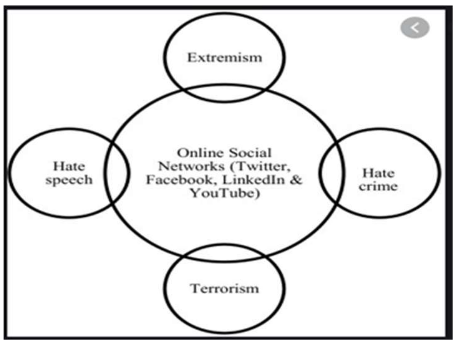

Fig. 1 Hate speech review in social media

Currently, online hate speeches are growing day by day making their automatic detections compulsory. Studies on security issues with social media 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. These networks do not concentrate on removing these kinds of speeches or words. One alternative however is the automation of tasks that 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 researchers [5] [6]. Despite improvements in the field, issues arise in the 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 Social Media Analytics (SMAs) are N-grams, Bag of words, Term Frequency-Inverse Document Frequency (TFIDF), 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. (Support Vector Machine (SVM) 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

Using a dataset, this study aims to solve the challenge of identifying hate speech on Twitter. The purpose of this work is to enhance hate speech detections from datasets by using MPCA and ECNN. The next section is a

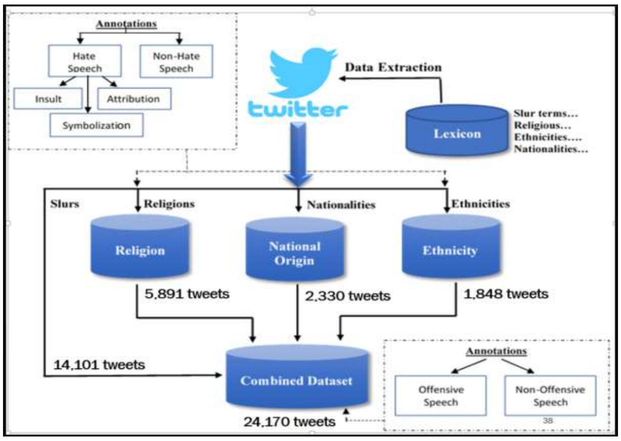

Fig. 2 Hate speech detection Model

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 were 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 with their techniques in Ethiopian, Amharic texts. A customized dataset was created by crawling Facebook 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.

Crowdsourcing was used in [12] where the authors used it to collect hate speech tweets. The crowd-sourced 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 coordinates 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 coordinate 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. The method, which was given the name PDCA (Principal Component Discriminant Analysis), was found to increase classifier accuracy over PCA and LDA. The study's experiments on an urban and agriculture image showed its efficiency. Liu et al. (2019) [25] also voted LDA to detect the hate speech. They used multi-task learning technique.

A supervised approach for extracting n-grams from tweets was proposed in [15]. A Twitter-specific vocabulary for SA (Sentiment Analysis) was created with the help of this study's feature reduction techniques (n-grams and statistical analysis). 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 [27].

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. Many applications, including content-based image searching and retrieval, have benefited from training these neural networks over large imagery datasets because the fidelity provided by the convolution filters of CNNs can promote accurate content searches within large datasets [28]. The filters used in convolution are automatically selected during the operation of convolution. The majority of convolution filters include Soble filter, median filter and average filters. The detailed description for filtering can be found in [26].

3. 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 the overall performance of hate speeches. This proposed work used uses MPCA algorithm for extracting important features which are then classified by the proposed ECNN.

3.1 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 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. These works use 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 semantically similar words. 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 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 the NLTK library [18] for removing stop words and to reduce the impact of stop words in Twitter sentiment classification. Unigram method is used for the process of tokenization. Normalization is the final step of pre-processing in this work. This work uses Min-Mix Normalization, a technique that linearly transforms data [19] and fits it into a pre-defined boundary. This normalization can be depicted as Equation (1)

$$A' = \left(\frac{A - \min valueof}{\max valueofA - \min valueofA)}\right) * (D - C) + C \quad (1)$$

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

3.2 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 ussie 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 jth element of the ith feature vector, then SD √ (λ _j) can be applied to the feature vector. The resulting feature vector y_i^' can be rewritten as Equation (2)

$$y_i' = \left[\frac{y_{i0}}{\lambda_0}, \frac{y_{i1}}{\lambda_1}, \ldots \frac{y_{i(r-1)}}{\lambda_{r-1}}\right] \quad (2)$$

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)

$$Y = TX \quad (3)$$

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

$$\lambda I - S)U = 0 \quad (4)$$

Where the square matrices: I - has identity values in their diagonal; S - covariance matrix of original data, U and λ –eigenvectors. Eigenvalues U_j and λ_j (j=1,2,…m) is computed using (2) in the order λ_1≥λ_2≥⋯≥λ_m. Therefore, the eigenvector U of the covariance matrix define the lower dimensions as U= 〖[U〗_1,U_2,……,U_m].

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

$$Y = T'X \quad (5)$$

$$V_N = b_1u_1 + b_2u_2 + \cdots . + b_Nu_N \quad (6)$$

$$S = \sum_{i=0}^{1} b_1u_1; 1 < N \quad (7)$$

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 Eigenvectors 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., $C = AA^T$ Eigenvalues are obtained from the covariance matrixes that are $V_1V_2V_3V_4 \ldots V_N$
6) Finally, Eigenvectors are calculated for covariance matrix C
7) Any vector S can be written as a linear combination of Eigenvectors using (6)
8) Only the Largest eigenvalues 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

3.3 ECNN algorithm for detecting hate speech detection

In order to divide test data into yes/no categories, this work employs an improved convolutional neural network (CNN) dubbed ECNN. The employed CNN has many hidden layers, including convolution, pooling, and fully connected layers, as well as input and output layers. The inputs are processed in a convolution layer (which mimics a neuron's response) before being passed on to the next layer. Figure 3: Schematic representation of the ECNN architecture.

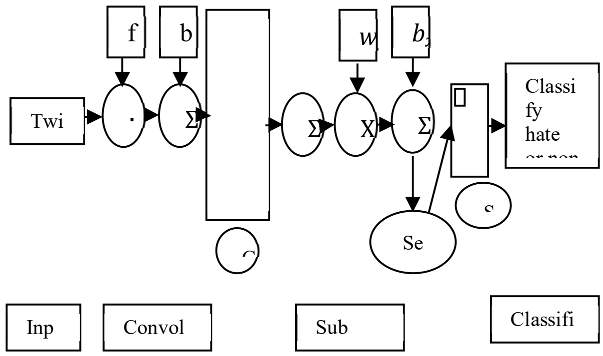

Fig 3 Architecture diagram of ECNN

Local or global pooling layers can be used in CNNs to combine the results of a neuron's clusters with the results of the next neuron. Mean pooling implements average values from neuron clusters in the preceding layer, whereas Fully linked layers map each neuron to neurons in other layers. Conventional MLP (Multi-Layer Perceptron) neural networks [22] are the conceptual ancestors of CNNs. The ECNN used in this study has input, convolution, subsampling, and classification layers, allowing for effective analysis of data in high dimensions. Convolution layers' shared parameters make it possible to use fewer of them.

The input layer of an ECNN takes raw data (Tweets) and transforms and passes it on to the next layer. This layer also defines initial parameters like the scale and filters for local fields. Convolution layer (Cx) combines inputs from previous convolution layers with new inputs to generate many layers of data (feature maps). This layer is responsible for extracting important features and reducing the network's computational complexity.

In the forward and backward pass, during the training of neural network, it is assumed that error should me minimum. Its failures justified by the improper selection of weights. Backpropagation algorithm is the most suitable candidate in selecting the weights for minimizing the error. There are various significant points are noticed like memory significant in case of large network and lack of parameter tuning making its overhead-free.

Within the convolution layer, an activation function is executed. This function creates a non-linear network structure by mapping outputs to inputs. Backpropagation is the usual and efficient method for assigning and updating feature value weights. This procedure computes the gradient activation function for the specified NN weight. The addition of weights to feature values for a new pattern output defined by Equations (8) and (9)

$$y(n) = f(\sum_{i=1}^{i=N} w_i(n)x_i(n) \ ) \quad (8)$$

$$Where\ f(x) = \begin{cases} +1 & if x \geq 0 \\ -1 & if x < 0 \end{cases} \quad (9)$$

Where n stands for the iteration index, the Connection weights are then updated based on (10)

$$w_i(n+1) = w_i(n) + \eta(d(n) - y(n)x_i(n) \ ), \quad i = 1,2,..N \ \ (10)$$

Where, η is the gain factor

And the application of SD is defined in Equation (11)

$$\sigma = \sqrt{\frac{1}{n} f_i(x_i - \overline{x}))^{\frac{1}{2}}} \ \ (11)$$

When fed the weighted features, ECNN can make more informed classifications. When using a polynomial distribution function, you may rest assured that you're always looking at the same data. In this layer, sub-samples are taken from each feature map generated by the previous convolution layer. Using a genetic fitness value for feature selections, ECNN can accurately categorize tweets as either hostile or neutral. The suggested CNN benefits from the use of genetic fitness values and polynomial distribution. The genetic operator chooses two samples to serve as parents for the offspring. Iterations of this process are taken until the individuals with the highest fitness levels are identified. The selection operator for random selections can be defined as Equation (12)

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

Where, P(c_i) - distribution of chromosome c_i in a population of size n. This means that chromosomes with higher fitness scores are chosen, and consequently, children born after a crossover operation have increased fitness.

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. The overall workflow of the proposed system is depicted in Figure 4.

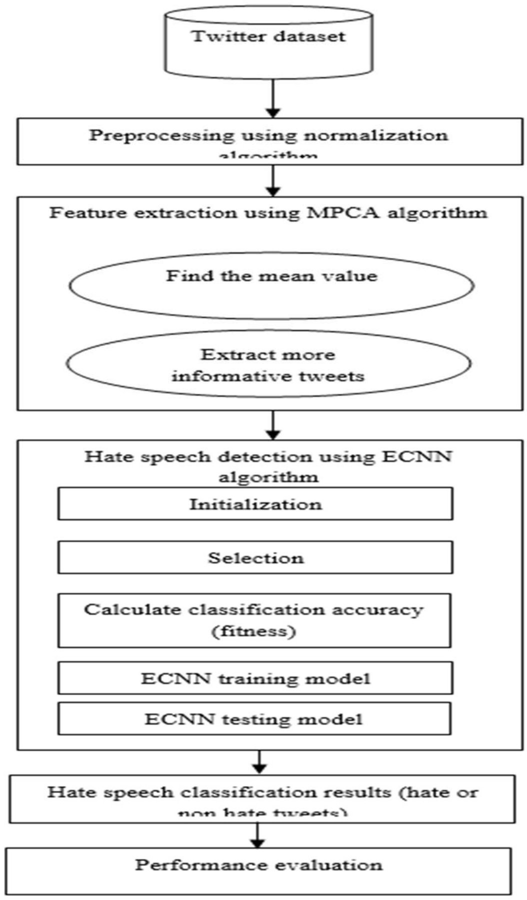

Fig 4 Overall block diagram of the proposed system

4. Experimental Result

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) the number of tweets where sixteen percent are hate speeches, thirty-three percent are offensive without hate speech while the balance is 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 measures how close a model comes to its predictions in terms of the proportion of true positives. High False Positive Costs Can Be Identified Through Measures of Accuracy.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall: The percentage of True Positives that a model correctly identifies is known as its Recall. When the cost of a False Negative is high, Recall should be used as the metric to choose the best model.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

F1 Score: A test's reliability is quantified by its F-measure.

$$F_1\text{-score} \quad \frac{1}{\frac{1}{2}\left(\frac{1}{\text{recall}} + \frac{1}{\text{precision}}\right)}$$

Accuracy: It measures how well an algorithm performs on a set of test tuples.

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

The comparative performance of classification methods for precision is depicted in Figure 5.

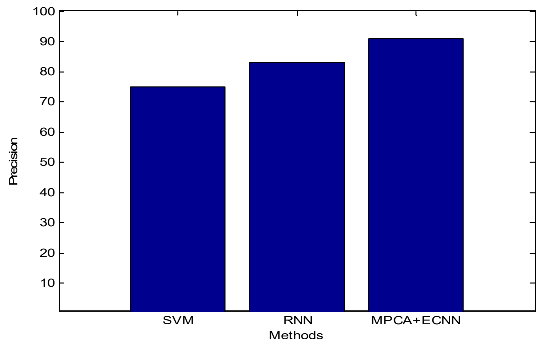

Fig 5 Precision

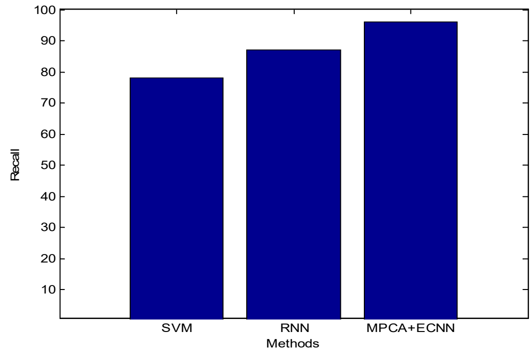

Fig 6 Recall

Compared to MPCA+ECNN, SVM and RNN have lower precision values in the above figure, suggesting that the proposed technique accurately recognizes hate speech. Figure 6 depicts the relative recall performance of various classification approaches. As can be seen in the above graph, MPCA+ECNN has higher recall values than SVM or RNN algorithms, suggesting that it is more accurate at identifying hate speech.

. Comparative performance of classification methods with respect to F-measure is depicted in Figure 7.

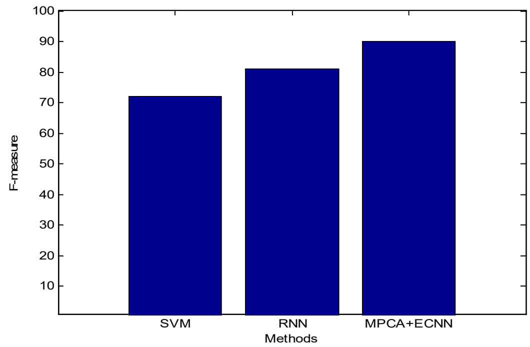

Fig 7 F-measure

It is evident from the above figure that SVM or RNN algorithms have lower F-measure values when compared with MPCA+ECNN implying that with the proposed method, hate speech can be identified with a higher recall rate. Figure 8 shows how well different classification strategies fare in terms of Accuracy.

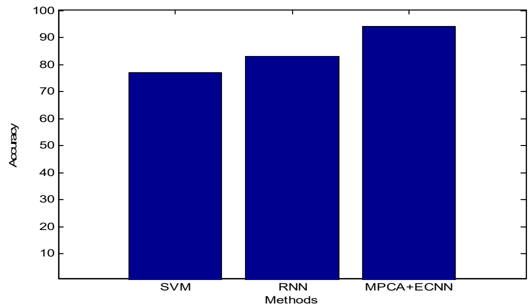

Fig 8 Accuracy

Table 1: Overall Summary of the performance measures for hate Speech Detection

Methodology Performance Measures	*SVM*	*RNN*	*MPCA+ECNN*
Recall (%)	79.41	86.35	91.37
F-Measure (%)	71.28	81.05	89.16
Accuracy (%)	79.50	84.80	95.20

As can be seen in the figure above, the F-measure values for SVM or RNN algorithms are lower than those for MPCA+ECNN, suggesting that the suggested method is more effective at identifying hate speech due to higher recall values. On the Twitter hate speech dataset, the recommended MPCA+ECNN approach stands head and shoulders above the competition. Table 1 displays the overall accuracy of the proposed model in detecting hate speech. We found that the best outcomes could be achieved by combining MPCA with ECNN as opposed to any of the other approaches we tried.

5. Conclusion

To summarize the proposed work pre-processes, the hate speech dataset by changing tweet text into lower case and cleans the dataset by eliminating URLs, white spaces, usernames, hashtags, stop-words and punctuations. The tweets are then tokenized using unigram method 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 significance of the proposed work lies in using the modified PCA which is a mathematical procedure that maps high dimensional data to lower-dimensional data using a linear transformation where lower dimensions are defined by Eigenvectors of the covariance matrix. Thus, MPCA extracts hate speech features with reduced errors. 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. The future scope would be to propose optimizations for handling voluminous datasets using fuzzy clustering.

References

[1] H. Watanabe, M. Bouazizi, T. Ohtsuki, Hate speech on twitter a pragmaticapproach to collect hateful and offensive expressions and perform hate speech detection, IEEEAccess, 13825-13835, 2018.

[2] Z. Zhang, D. Robinson, J. Tepper, Detecting hate speech on twitter using aconvolution-gru based deep neural network, In15th European Semantic Web Conference, pp745–760, 2018

[3] S. Malmasi, M. Zampieri, Detecting hate speech in so-cial 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] P. Fortuna, Automatic detection of hate speech in text: anoverview of the topic and dataset annotation with hierarchical classes, 2017.

[6] D. Robinson, Z. Zhang, J. Tepper, Hate speech detection on Twitter: feature engineering vs feature selection, European Semantic Web Conference, pp 46-49, 2018.

[7] R. Gomez, J. Gibert, L. Gomez, D. Karatzas, Exploring hate speech detection in multimodal publications, Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020.

[8] Shirbhate G. Amit, S N. Deshmukh, Feature Extraction for Sentiment Classification on Twitter Data, International Journal of Science and Research (IJSR), pp.2183-2189, 2016
[9] R. Batool, A.M Khattak, J. Maqbool, S. Lee, Precise tweet classification and sentiment analysis. 2013 IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS). IEEE, 2013.
[10] M. Ahmad, S. Aftab, I.Ali. Sentiment analysis of tweets using svm, Int. J. Comput. Appl, pp 25-29, 2017
[11] Z. Mossie, J.H. Wang, Vulnerable community identification using hate speech detection on social media, Information Processing & Management, p102087, 2020.
[12] T. Davidson, D. , Warmsley, M. Macy, Weber, Automated hate speech detection and the problem of offensive language, Proceedings of the International AAAI Conference on Web and Social Media. pp 512-515, 2017.
[13] G. Haibo, H. Wenxue, C. Jianxin, X. Yonghong, Optimization of principal component analysis in feature extraction, International Conference on Mechatronics and Automation, IEEE, pp 3128-3132, 2007.
[14] M. Imani, H. Ghassemian, Principal component discriminant analysis for feature extraction and classification of hyperspectral images, Iranian Conference on Intelligent Systems (ICIS), pp 1-5, 2014.
[15] M. Ghiassi, J. Skinner, D. Zimbra, Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network, Expert Systems with applications, pp 6266-6282. 2013
[16] J. Freeman, Content search within large environmental datasets using a convolution neural network, Computers & Geosciences, p 104479, 2020
[17] B. Riordan, M.N. Jones, Redundancy in perceptual and linguistic experience: Comparing feature-based and distributional models of semantic representation, Topics in Cognitive Science, pp 303-345, 2011
[18] A. Farkiya, P. Saini, S. Sinha, S. Desai, Natural Language Processing using NLTK and WordNet, pp 5465-5469,2015.
[19] S. Jain, S. Shukla, R. Wadhvani, Dynamic selection of normalization techniques using data complexity measures, Expert Systems with Applications, pp 252-262, 2018
[20] Y.H. Taguchi, Y. Murakami, Principal component analysis based feature extraction approach to identify circulating microRNA biomarkers, PloS one, pe66714, 2013
[21] A. Pal, Principal Component Analysis of TF-IDF In Click Through Rate Prediction‖, International Journal of New Technology and Research (IJNTR), pp 24-26, 2018.
[22] M. Suganuma,, S. Shirakawa, N. Nagao, A genetic 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
[24] H. Almerekhi, H. Kwak, B.J. Jansen, J. Salminen, Detecting toxicity triggers in online discussions, In: The proceedings of the 30th ACM conference on hypertext and social media, pp 291-292, 2019.
[25] H. Liu, P. Burnap, W. Alorainy, M.L. Williams, Fuzzy multi-task learning for hate speech type identification, In The world wide web conference, pp. 3006-3012, 2019
[26] J.S. Lee, Refined filtering of image noise using local statistics, Computer graphics and image processing, pp 380-389, 1981.
[27] N. Kumar, A. Sharma, Sentimental analysis for political activities from social media data analytics, 2017.
[28] N. Kumar, N. Sukavanam, A cascaded CNN model for multiple human tracking and re-localization in complex video sequences with large displacement. Multimedia Tools and Applications, pp 6109-6134, 2020.

Majed Alowaidi received the B.Eng. degree (Hons.) from the Riyadh College of Technology, in 2006, and the M.A.S. and Ph.D. degrees in computer engineering from the University of Ottawa, in 2012 and 2018, respectively. Since 2012, he has been a member of MCRLAB, University of Ottawa, Canada. He is currently working as an Assistant Professor and the Head of the IT Department, College of Computer and Information Science, Majmaah University, Al Majma'ah, Saudi Arabia. His research interests include cybersecurity, the IoT, semantic web, cloud and edge computing, and smart city.