# HATE SPEECH DETECTION FROM SOCIAL MEDIA USING ONE DIMENSIONAL CNN COUPLED WITH GLOBAL VECTOR

**Subhajeet Das[1,*], Koushikk Bhattacharyya[2,] And Sonali Sarkar[3]**

## Abstract

Hate speech is used to hurt someone or a community by spreading it over social media platforms. Every day users scroll through the posts on social media and like, comment, and share these posts. But before commenting, they generally don’t think about whether it is going to hurt someone or a group of people or not. Sometimes this happens intentionally and sometimes unintentionally. But the effect of it is unchangeable. It can’t be stopped from happening, but this can be detected whether it is actually intended to hurt or spread hatred within society. This research article easily detects whether it is hate speech or not by going through the comments collected over time using 1 Dimensional Convolutional Neural Network (1D-CNN) coupled with 840 Billion Global Vector (GloVe) classifier.

**Keyword:-** Convolutional Neural Network, Word Vector, Global Vector, Confusion Matrix, Precision, Recall, Accuracy.

[1*]Department of Computer Science & Engineering, Swami Vivekananda Institute of Science & Technology, Kolkata, India. email : subhajeetdas.official@gmail.com
[2]Department of Computer Science & Engineering, Swami Vivekananda Institute of Science & Technology, Kolkata, India. email : koushikkbhattacharyya@gmail.com
[3]Department of Chemical Engineering, Swami Vivekananda Institute of Science & Technology, Kolkata, India. email : drsonalisarkar30@gmail.com

***Corresponding Author:-** Subhajeet Das
[*]Department of Computer Science & Engineering, Swami Vivekananda Institute of Science & Technology, Kolkata, India. email : subhajeetdas.official@gmail.com

**DOI: -** 10.48047/ecb/2023.12.si10.0081

## 1. Introduction

Nowadays with the increase in smartphone and computer usage, the usage of social media like-Facebook, WhatsApp, Twitter, LinkedIn, Instagram, etc. has also increased a lot over the past few years. This also elevates the hateful and malicious activities in this social media. These activities may include cyberbullying, usage of abusive language, posting explicit content like pictures and videos, promoting racism, hateful and offensive speech to any individual (it can be anyone starting from normal human beings, politicians, sports person, government officials, actors, celebrities, certain products and so on) or a group (LGBTQ, gender, certain religion, specific community, organizations, countries, etc.). These things are done intentionally or unintentionally, but the effect is inevitable. This draws a major negative impact on the popularity and trustworthiness of the social platform. This impacts the freedom of speech of humans, which is one of the important fundamental rights. It also discourages good deeds and affects mental health. But from all the above-mentioned social media, Twitter, Facebook and Instagram are mostly affected by these malicious activities.

Although the hateful comments and contents can be removed from social media, but it is a hectic job and requires human intervention throughout the process to understand the mentality and intention behind that post. This manual process is non-scalable and the complexity is also very high as the natural language construction changes based on the different types of targets, different types of products, and even it changes from one region to another region while keeping the inner meaning the same. So, except for this manual sorting out of the contents, an automated algorithm is required to ease the process. Most of the previous works on these include the usage of either extracting the features manually [17] or using representation learning methods followed by linear classifiers [14,18]. But the recent advancements in the Deep Learning method show us the improvement of accuracy over many complex problems of vision, speech, and text.

In this research paper, we are doing the experimentation with 1-D Convolutional Neural Network incorporated with Global Vector and a best-in-class annotated social media Comments dataset and thereby achieving a state-of-art accuracy at detecting the hate comments from social media.

## 2. Literature Study

Using the Continuous Bag-of-Word neural language model, Nemanja Djuric et al. proposed to use Paragraph2vec to learn the distribution of comments and words in a shared space [18]. Author Armand Joulin et al. proposed the use of logistic regression or Support Vector Machine on Bag-of-Word (BoW) for sentence classification, which results in a FastText classifier that can perform well in standard multi-core CPU [9]. Convolutional Neural Networks (CNNs) that are trained on the pre-learned word vector for sentence classification and then utilized for sentiment analysis were the subject of substantial research by Yoon Kim et al [11]. Zeerak Waseem et al. used the character n-gram for detecting the hate speech and also enlisted the concerns observed from Critical Race Theory [17]. A machine learning algorithm created by Chikashi Nobata et al. can detect hate speech in online user comments and also created an annotated user comment dataset of abusive language [14]. Samuel Brody et al. introduced an automatic method that facilitates word lengthening to understand the hidden sentiments in the comments on many social media, and microblogs [1]. John A. Bullinaria et al. proposed a structured finding of principal computational probabilities for articulation and authentication of the inner meaning of words from the co-occurrence statistics [2]. Yunfei Chen et al. tried to explore the 4I’s namely Identity, Inference, Influence, and Intervention of cyberbullying, and automatically detect them in real-time on social media using a machine learning approach [3]. Ronan Collobert et al. proposed a single CNN model that can predict part of speech, tags, chunks, semantic roles, and similarities between words using a language model, given a sentence, incorporating both multitask learning and semi-supervised learning for improvement [4]. Li Dong et al. developed a statistical parser that can be used to classify the sentiment at the sentence level, using Context Free Grammar (CFG) [5]. Stephan Gouws et al. found out the distinction between the volume of abridged English terms utilized by different groups of users to convey themselves in social media [6]. Minqing Hu et al. focused on the procedure to mine and summarizing customer reviews whether they were positive or negative on a particular product [7].

The procedure has mainly three steps for detection:
Step 1: Mining the features of the product from the customer comments.

Step 2: Identifying the judgment sentence whether the review was positive or negative.
Step 3: summarizing the results.

Eric H. Huang et al. discussed about a new neural network architecture capable of learning word embeddings from the semantics using local and global document contexts and capturing the similarities in meaning or having more than one meaning in the same word [8]. Kettrey et al. tried to analyze a certain amount of YouTube comments to find out the hidden patterns of racism of many kinds using Logistic Regression [10]. Irene Kwok et al. discussed about a supervised machine learning approach to determine whether the comments from Twitter are "Racist" or "Nonracist" using Binary Classification and they achieved a total of 76% accuracy rate in detecting anti-black hate speech [12]. Tomas Mikolov et al. explored that the vector-space representation of words was very efficient at apprehending semantic and syntactic similarities in sentences by using the vector offset method. It was later used to answer SemEval-2012 Task 2 questions effectively [13]. Bo Pang et al. tried to find out the polarity of the sentiment by categorizing the text into smaller subjective segments by finding the minimum cuts in the graph to obtain the result [15]. Yelong Shen et al. discussed the application of CNNs on semantic models to find out a low-dimensional semantic vector in web documents. At first, the word N-gram model is generated using convolution max pooling. Then, the global feature vector is formed by combining the local features extracted from the sequence of the word. Ultimately, the higher dimensional semantic features are extracted from the global feature vector [16]. Author Subhajeet Das et al. executed an extended research on different ML algorithms and found that out of all models, Random Forest and Decision tree outperformed [19].

## 3. Proposed Method & Algorithm

***Step 1:*** *Collecting the appropriate data sets (In our case, we've got both the train and test data).*
***Step 2:*** *Cleaning and analyzing the data.*
***Step 3:*** *Up-Sampling and Tokenizing the cleaned comments with maximum feature size of 20,000.*
***Step 4:*** *Creating the model using the 1D-CNN classifier coupled with 840 Billion Global Vectors having 2 Convolutions, 2 Dense Layers, 2 Dropout Layers, 1 One-Dimensional Max Pooling Layer & 1 One-Dimensional Global Max Pooling Layer.*
***Step 5:*** *Training the Model in 3 epochs having 32 batch size and validating the model.*
***Step 6:*** *Generating the Precision score, Recall score, Accuracy Score, F1 score, Confusion matrix, Accuracy (Training & Validation Accuracy) vs. Epochs curve, and Loss (Training & Validation Loss) vs. Epochs curve to check the model's accuracy.*
***Step 7:*** *Feed the model with actual data samples collected from Twitter, Facebook, and Instagram to detect hate speech.*

### *3.1 Flow Chart*

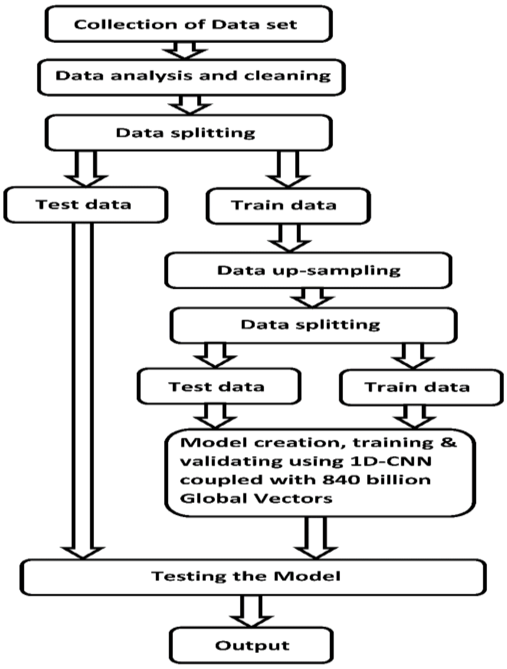

***Figure 1.*** *The working of the Proposed Method*

## 4. Applied Proposed Method

*Step 1: Collecting the appropriate data sets (In our case, we've got both the train and test data).*

A. Appropriate datasets are being collected.

B. In case of Twitter, the Training dataset contains a total of 1,43,742 annotated data and the Testing dataset contains 65,103 data.

C. In case of Facebook, the Training dataset contains a total of 3,03,651 annotated data and the Testing dataset contains 1,31,000 data.

D. In case of Instagram, the Training dataset contains a total of 1,88,218 annotated data and the Testing dataset contains 81,000 data.

*Step 2: Cleaning and analyzing the data.*

A. Any fields with missing values and special characters are removed.

B. All the text contents are converted to its lower-case.

*Step 3: Up-Sampling and Tokenizing the cleaned comments with maximum feature size of 20,000.*

A. After up-sampling the cleaned data, for Twitter the positive data (label '1') count is 1,30,760 and negative data (label '0') is 1,30,760, in total 2,61,520. For Facebook, the positive data (label '1') count is 2,19,107 and negative data (label '0') is 2,19,107, in total 4,38,214. For Instagram, the positive data (label '1') count is 1,23,740 and negative data (label '0') is 1,23,740, in total 2,47,480.

B. The words used in hate speech and non-hate speech are visualized using WordCloud.

C. The difference between the numbers of initial original data and up-sampled data is shown in a histogram format using Seaborn and Matplotlib.

D. Tokenizing the data using TensorFlow's Keras neural network library's text.Tokenizer with maximum feature size of 20,000 and further converting it into a 2D Numpy array of integers.

*Step 4: Creating the model using the 1D-CNN classifier coupled with 840 billion Global Vectors having 2 Convolutions, 2 Dense Layers, 2 Dropout Layers, 1 One-Dimensional Max Pooling Layer & 1 One-Dimensional Global Max Pooling Layer.*

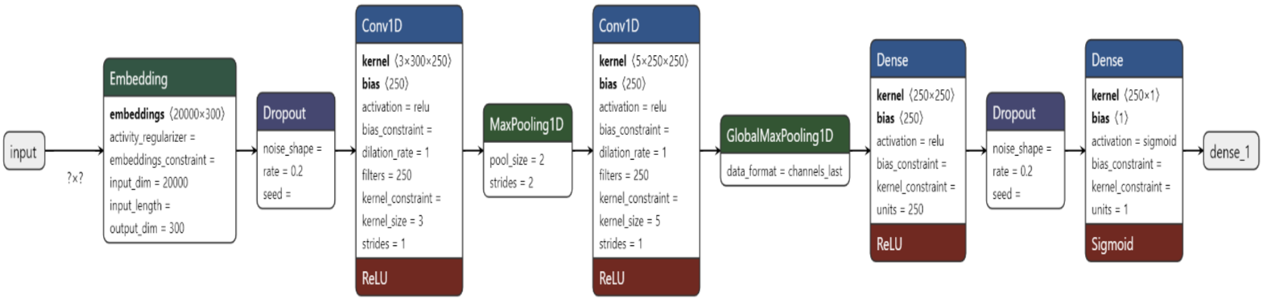

***Figure 2.*** *Layers of the 1D-CNN with 840 billion Global Vector Embedding*

*Step 5: Training the Model in 3 epochs having 32 batch size and validating the model.*

*Step 6: Generating the Precision score, Recall score, Accuracy Score, F1 score, Confusion matrix, Accuracy (Training & Validation Accuracy) vs. Epochs curve, and Loss (Training & Validation Loss) vs. Epochs curve to check the model's accuracy.*

*Step 7: Feed the model with actual data samples collected from Twitter, Facebook, and Instagram to detect hate speech.*

## 5. Results & Observations

### *5.1. Confusion Matrix*

The confusion matrix (Figure 3, 4 and 5) is the performance metric that is used for evaluating the classification model. It is often denoted as a table composed of 4 cells establishing the relation between actual values and predicted values. These contents of the cells are described as:

A. True Positive (TP): True Positive data is what happens when a model predicts something positive and it actually turns out to be positive. Here the number of True Positive values for Twitter, Facebook and Instagram are 27404, 69746, 36201 respectively.

B. True Negative (TN): True Negative value is what happens when a model predicts something negative and it actually happens to be negative. The number of True Negative values are 30266, 54859, 37845 respectively.

C. False Positive (FP): if the model predicts positive but it is labeled as negative in the actual data, then it is known as False Positive or Type 1 Error value. From the confusion matrix, the number of False Positive values are 1409, 3204, 3929 respectively.

D. False Negative (FN): False Negative or Type 2 Error Value is used to describe when a model predicts something negative but the actual data is categorized as positive. The total number of False Negative values are 3021, 3191, 3205 respectively.

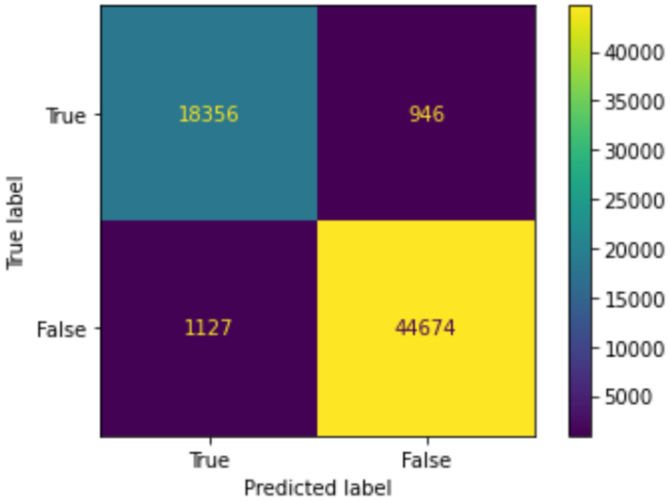

***Figure 3.*** *Confusion Matrix for Twitter data-set*

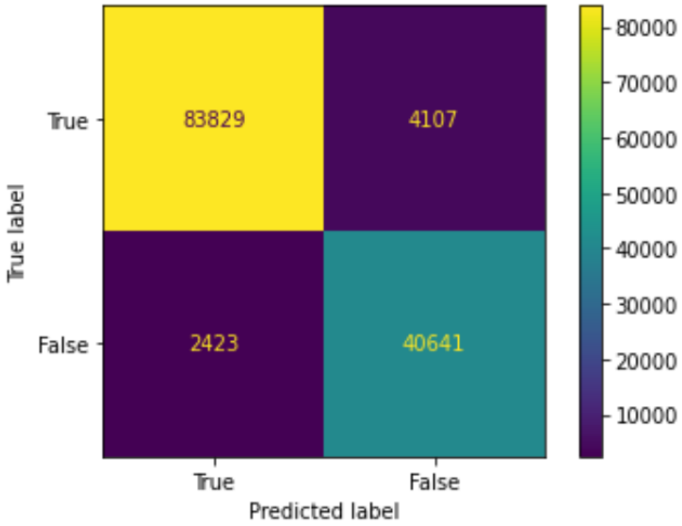

***Figure 4.*** *Confusion Matrix for Facebook data-set*

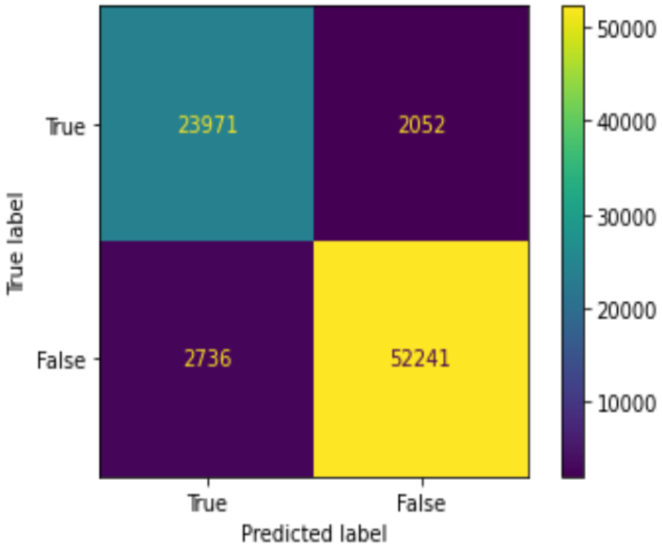

***Figure 5.*** *Confusion Matrix for Instagram data-set*

### *5.2. Accuracy vs. Epochs Curve*

The Accuracy versus Epochs curve displays the ratio of Training and Validation Accuracy plotted in Y and The number of Epochs plotted across X axis respectively. In Figure 6, the Blue line represents the Training Accuracy whereas, the Red line represents the Validation Accuracy of the 1-D CNN classifier. From the figure it is clearly evident that the Training Accuracy improves over time while the Validation Accuracy decreases as the number of Epochs increases.

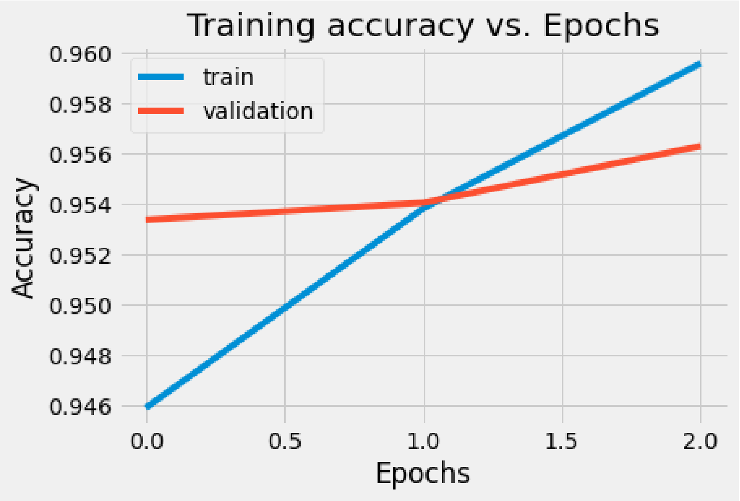

***Figure 6.*** *Precision-Recall Curve*

### *5.3. Loss vs. Epochs Curve*

The ratio of Training Loss and Validation Loss versus number of Epochs are shown in the Loss vs. Epochs curve in Y and X axis respectively in Figure 7. It is very useful in evaluating the performance of the model. The Training Loss decreases as the number of Epochs increases, while the Validation Loss increases at first but then it gradually decreases.

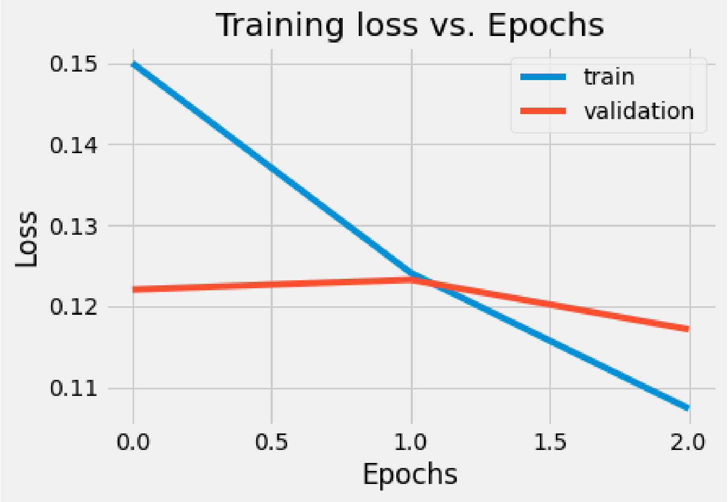

***Figure 7.*** *ROC Curve*

## 6. Performance Measurement

There are certain measures, that need to be taken to evaluate the efficiency or how much accurate the model is at predicting the outcome. The following are some measures, used for analyzing the performance.

### *A. Precision*

According to its definition, it is the proportion of all TP values to all Predicted Positive values i.e., the sum of the TP and FP values. Precision depicts how much the model is precise at predicting the actual positive values out of the predicted positive values.

$$Precision = \frac{TP}{(TP+FP)} \qquad (1)$$

### *B. Recall*

It is referred to as the ratio of TP values to all Actual Positive values i.e., the total of TP and FP values. Recall describes how much the model is able to predict values that are positive out of all the positive values. Recall is directly proportional to the rate of positive samples detected. It is also called Sensitivity or True Positive Rate (TPR).

$$Recall\ or, Sensitivity\ or, TPR = \frac{TP}{(TP+FN)} \qquad (2)$$

### *C. Accuracy*

It is the ratio of accurate predictions to all predictions. It is the metric that describes how the model is performing over all the classes when all the classes have equal significance.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \qquad (3)$$

### *D. F1 Score*

It is the harmonic mean of a model's recall and precision. The main purpose of the F1 score is to merge both precision and recall metrics into a single metric. It measures the selected model's accuracy on a particular data-set.

$$F1\ Score = 2 \times \frac{(Precision \times Recall)}{(Precision+Recall)} \qquad (4)$$

### *E. Specificity*

It is defined as the accurate identification of all actual negative values that are negative. A higher value of specificity shows a better performance of the model. It is also called as True Negative Rate (TNR).

$$Specificity\ or, TNR = \frac{TN}{(TN+FP)} \qquad (5)$$

### *F. False Positive Rate (FPR)*

False Positive Rate describes out of all the negative values, how many values are incorrectly classified as positive i.e., FP.

$$FPR = 1 - Specificity = \frac{FP}{(FP+TN)} \qquad (6)$$

### *G. False Negative Rate (FNR)*

False Negative Rate describes the ratio of the predicted FN values out of all actual positive values i.e., the summation of TP and FN values. It is denoted as the negation of the True Positive Rate (TPR) from 1.

$$FNR = \frac{FN}{(TP+FN)} \qquad (7)$$

### *H. False Discovery Rate (FDR)*

It is represented as the ratio of FP values to all predicted positive values.

$$FDR = \frac{FP}{(FP+TP)} \qquad (8)$$

### *I. Positive Predictive Value (PPV)*

It is mainly used for analyzing the model statistically. It is similar to precision but the main difference is that precision is expressed in the probabilistic form, whereas PPV is expressed in the percentage form.

$$PPV = \frac{TP}{(TP+FP)} \times 100 = Precision \times 100 \qquad (9)$$

### *J. Negative Predictive Value (NPV)*

It is the ratio of TN values out of all predicted negative values i.e., the summation of TN and FN. It is also used for statistically analyzing the model and expressed in percentage form.

$$NPV = \frac{TN}{(TN+FN)} \times 100 \qquad (10)$$

| Experiment | Result |
|---|---|
| Precision | 0.942154698968 |
| Recall /Sensitivity /True Positive Rate | 0.950989534763 |
| Accuracy | 0.968158149394 |
| F1 Score | 0.946551501869 |
| Specificity / True Negative Rate | 0.975393550359 |
| FPR | 0.024606449641 |
| FNR | 0.049010465237 |
| FDR | 0.057845301032 |
| PPV | 94.215469896833 |
| NPV | 97.926348092942 |

***Table 1.*** *The observed results after training the model with Twitter data*

| Experiment | Result |
|---|---|
| Precision | 0.971907897788 |
| Recall /Sensitivity /True Positive Rate | 0.953295578603 |
| Accuracy | 0.950152671756 |
| F1 Score | 0.962511768893 |
| Specificity / True Negative Rate | 0.943734906186 |
| FPR | 0.056265093814 |
| FNR | 0.046704421397 |
| FDR | 0.028092102212 |
| PPV | 97.190789778788 |
| NPV | 90.821936175918 |

***Table 2.*** *The observed results after training the model with Facebook data*

| Experiment | Result |
|---|---|
| Precision | 0.897554948141 |
| Recall /Sensitivity /True Positive Rate | 0.921146677939 |
| Accuracy | 0.940888888889 |
| F1 Score | 0.909197800114 |
| Specificity / True Negative Rate | 0.950233734107 |
| FPR | 0.049766265893 |
| FNR | 0.078853322061 |
| FDR | 0.102445051859 |
| PPV | 89.755494814094 |
| NPV | 96.220507247712 |

***Table 3.*** *The observed results after training the model with Instagram data*

## 7. Conclusion

In this research paper, we tried to find out the application of the 1-Dimensional Convolutional Neural Network coupled with 840 billion Global Vectors in the field of detecting hate speech on social media. This research came up with a simple yet very efficient unique model that outperforms all the existing models. This model achieved a state-of-art 97% accuracy at Twitter, 95% accuracy at Facebook and 94% accuracy at Instagram in determining whether a comment is a hate speech or not. In the future, the main aim will be to make this model efficient enough to detect hate speech on more social media platforms in an automated manner.

## Reference

1. Brody, Samuel, and Nicholas Diakopoulos. 2011. Cooooooooooooooollllllllllllll!!!!!!!!!!!!!! using word lengthening to detect sentiment in microblogs. *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 562-570.
2. Bullinaria, John A., and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods* 39, no. 3 (2007): 510-526.
3. Chen, Yunfei, Lanbo Zhang, Aaron Michelony, and Yi Zhang. 2012. 4Is of social bully filtering: identity, inference, influence, and intervention. *In Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 2677-2679.
4. Collobert, Ronan, and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multi-task learning. *In Proceedings of the 25th international conference on Machine learning*, pp. 160-167.
5. Dong, Li, Furu Wei, Shujie Liu, Ming Zhou, and Ke Xu. 2015. A statistical parsing framework for sentiment classification. *Computational Linguistics 41*, no. 2 (2015): 293-336.
6. Gouws, Stephan, Donald Metzler, Congxing Cai, and Eduard Hovy. 2011. Contextual bearing on linguistic variation in social media. I*n Proceedings of the workshop on language in social media (LSM 2011)*, pp. 20-29.
7. Hu, Minqing, and Bing Liu. 2004. Mining and summarizing customer reviews. *In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168-177.
8. Huang, Eric H., Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global

context and multiple word prototypes. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 873-882.

9. Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint* arXiv:1607.01759 (2016).
10. Kettrey, Heather Hensman, and Whitney Nicole Laster. 2014. Staking territory in the "World White Web" an exploration of the roles of overt and color-blind racism in maintaining racial boundaries on a popular web site. *Social Currents 1*, no. 3 (2014): 257-274.
11. Yoon, K. 2014. Convolutional Neural Networks for Sentence Classification [OL]. arXiv Preprint (2014).
12. Kwok, Irene, and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. *In Twenty-seventh AAAI conference on artificial intelligence*.
13. Mikolov, Toma´s, Wentau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. *In Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pp. 746-751.
14. Nobata, Chikashi, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. *In Proceedings of the 25th international conference on world wide web*, pp. 145-153.
15. Pang, Bo, and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint* cs/0409058 (2004).
16. Shen, Yelong, Xiaodong He, Jianfeng Gao, Li Deng, and Gregoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. *In Proceedings of the 23rd international conference on world wide web*, pp. 373-374.
17. Waseem, Zeerak, and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. *In Proceedings of the NAACL student research workshop*, pp. 88-93.
18. Djuric, Nemanja, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic et. al. 2015. Hate speech detection with comment embeddings. *In Proceedings of the 24th international conference on world wide web*, pp. 29-30.
19. Subhajeet Das, Koushikk Bhattacharyya, Sonali Sarkar. March 2023. Performance Analysis of Logistic Regression, Naive Bayes, KNN, Decision Tree, Random Forest and SVM on Hate Speech Detection from Twitter. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 7, Issue 3, pp 24-28.