Deep Learning Based Fusion Approach for Hate Speech Detection

Pawase Shubhangi.D[1], Pawar Gauri.v[2], Sable Shweta.B[3], Shinde Rajashree.N[4] ,

Prof. Dr. Jondhale.S.R[5]

[1,2,3,4,5]*Department of E&TC Engineering Amrutvahini COE, Sangamner, India*

---***---

Abstract - Hate speech on social media has been more common in recent years, and this has drawn attention to it as a major issue across the globe. The scientific community has been interested in hate speech detection algorithms, which have received major investment from numerous governments and organisations. Although there is a wealth of literature on this topic, because each proposed solution has unique benefits and drawbacks, it is still challenging to evaluate how well it performs. A significant attempt is made to improve the classification results in general by combining the output from different classifiers. Embeddings from Language Models (ELMo), Bidirectional Encoder Representation from Transformers (BERT), and Convolutional Neural Network (CNN) are a few well-known machine learning techniques for text classification that we first examine and use.

Key Words: machine learning, threatening speech, filtering, identification, CNN, NLP.

1.INTRODUCTION *(Size 11, Times New roman)*

The limited knowledge of facts with the toxic nature of OSNs often translates into ignominy or financial loss or both for the victim. Unenthusiastic speeches are in the form such as hate speech, bullying, profanity, flaming, trolling, etc., On the other hand, public shaming, which is the condemnation of someone who violates accepted social norms to arouse feelings of guilt in him or her, has not attracted much attention from a computational perspective.

The majority of modern civilization is powered by machine learning, including social network content filtering, e-commerce website suggestions, and a growing number of consumer goods like cameras and smartphones. Machine-learning algorithms are used to choose relevant search results, recognize objects in photos, convert speech to text, match news articles, posts, or products with users' interests, and more. These applications are increasingly using a group of methods known as deep learning. The ability of traditional machine-learning approaches to analyze natural data in its raw form was limited. For decades, designing a feature extractor that converted the raw data (such as the pixel values of an image) was crucial to building a pattern-recognition or machine-learning system. This required meticulous engineering and extensive domain knowledge.

When given raw data, a machine can automatically learn the representations required for detection or classification using a set of techniques called representation learning. Deep-learning techniques are representation-learning techniques that use many levels of representation. They are created by combining straightforward but non-linear modules that each convert the representation at one level (beginning with the raw input) into a representation at a higher, marginally more abstract level. Very complex functions can be learned by composing enough of these changes. Higher layers of representation accentuate characteristics of the input that are crucial for discriminating in classification tasks and decrease irrelevant variations. For instance, an image is made up of an array of pixel values.

ConvNets have been successfully used since the early 2000s for the detection, segmentation, and recognition of objects and regions in images. All of these tasks required the segmentation of biological images, particularly for connectomics, and the identification of faces, text, pedestrians, and human bodies in real-world photographs. Labelled data was relatively available for all of these tasks. Face recognition has been a significant recent practical success for ConvNets. The ability to classify images at the pixel level is significant because it will be used in technology, such as autonomous mobile robots and self-driving cars. These ConvNet-based techniques are used by businesses like Mobileye and NVIDIA in their forthcoming vision systems for automobiles.

2. LITERATURE SURVEY

O. de Gibert, N. Perez, A. García-Pablos, and M. Cuadros [3] Any communication that disparages a target group of people based on a trait like race, color, ethnicity, gender, sexual orientation, nationality, religion, or another feature is usually referred to as hate speech. The volume of hate speech is continuously rising as a result of social media's enormous growth in user-generated content. Along with the phenomenon's effects on society, interest in online hate speech identification and, in particular, the automation of this activity, has risen steadily over the past few years. This study describes a dataset of hate speech that includes thousands of words that have been manually classified as containing or not hate speech. The phrases were taken from the white nationalist forum Stormfront. To complete the hand labelling, a unique annotation tool has been created.

T. Davidson, D. Bhattacharya, and I. Weber[4] Little thought is given to the potential biases of the technologies being created and used to detect abusive language. In five separate sets of Twitter data that have been annotated for hate speech and abusive language, we look at racial bias. We use these datasets to train classifiers, and we compare the predictions of these classifiers to tweets written in Standard American English and African-American English. As classifiers developed using these datasets have a tendency to do, they tend to predict that tweets written in African-American English are abusive at significantly higher rates. The results demonstrate evidence of systemic racial bias in all datasets. Therefore, if these abusive language recognition algorithms are implemented, they will disproportionately harm African-American social media users.

T. Davidson, D. Warmsley, M. Macy, and I. Weber[5] The ability to distinguish hate speech from other objectionable language is a major obstacle for automatic hate-speech detection on social media. Lexical detection techniques frequently produce inaccurate results since they label any communications containing specific phrases as hate speech, and supervised learning experiments in the past have been unable to distinguish between the two categories. To gather tweets containing hate speech keywords, we employed a crowd-sourced hate speech lexicon. A sample of these tweets is divided into three groups using crowdsourcing, according to whether they contain hate speech, just objectionable language, or neither.

E. Cambria[6] Understanding emotions is crucial for personal growth and development, and as such, it is a crucial component of replicating human intelligence. Emotion processing is crucial for the development of AI and crucial for the closely related problem of polarity identification. Both the scientific community, with its intriguing open problems, and the business world, with its remarkable implications in marketing and financial market forecasting, have grown interested in the possibility of automatically capturing the general public's perceptions about social events, political movements, marketing campaigns, and product preferences. As a result, the fields of emotional computing and sentiment analysis are now developing, utilizing information retrieval, multimodal signal processing, and human-computer interaction to extract people's sentiments from the vast amounts of online data.

P. Badjatiya, S. Gupta, M. Gupta, and V. Varma[7] For applications like controversial event extraction, creating AI chatterbots, content recommendation, and sentiment analysis, hate speech detection on Twitter is essential. The ability to categorize a tweet as racist, sexist, or neither is how we describe this task. This assignment is extremely difficult due to the intricacy of the natural language constructs. To manage this complexity, we conduct extensive experiments with a variety of deep learning systems. Our tests on a benchmark dataset of 16K annotated tweets demonstrate that these deep learning techniques perform about 18 F1 points better than the most advanced char/word n-gram techniques.

3. PROPOSED WORK

The proposed approach for the purpose of achieving the hate speech recognition through the use of Neural Network given below using fig.1 In given figure we take input as tweet in text form from dataset and its tag name or keyword. Then we preprocess that data, extract their features and do its classification using CNN (Convolutional Neural Network) algorithm.
At the last stage of this algorithm technique as system trained and extraction of feature done result will be come is given input that is tweet is shaming or non-shaming with its admin name given below the

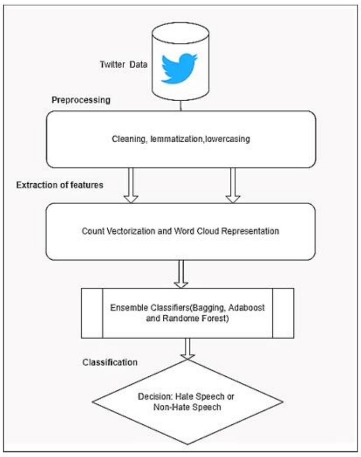

Twitter Data
Preprocessing
Cleaning, lemmatization,lowercasing
Extraction of features
Count Vectorization and Word Cloud Representation
Ensemble Classifiers(Bagging, Adaboost and Randome Forest)
Classification
Decision: Hate Speech or Non-Hate Speech

fig.1 System Architecture

3.1 CNN ALGORITHM

CNN stands for Convolutional Neural Network, which is a type of deep learning algorithm commonly used for image and video recognition, analysis, and processing. The basic architecture of a CNN includes convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply a set of filters to the input image to extract relevant features, while pooling layers down sample the feature maps to reduce their size and increase their computational efficiency. Fully connected layers then use the extracted features to classify the image into one or more categorize. CNNs are trained using large datasets of labeled images and use backpropagation to update the weights of the network to minimize
the difference between the predicted an actual output Some popular applications of CNNs include image classification, object detection, facial recognition, and autonomous driving

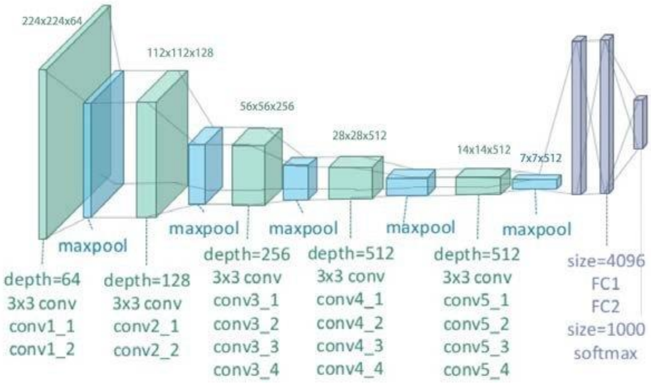


Fig.2 CNN Algorithm

3.2 Convolutional Layer

The first layer of a CNN is typically a convolutional layer. In this layer, the network applies a set of filters to the input image, each of which captures different patterns or features. The filters are small matrices of weights that are convolved with the input image to produce a set of feature maps. The size of the filters and the number of filters can be adjusted depending on the specific task. After the convolutional layer, the output is typically passed through a non-linear activation function, such as ReLU (Rectified Linear Unit), which introduces non-linearity into the network and helps to improve its performance.

3.3 Pooling Layer

A pooling layer is a type of layer commonly used in convolutional neural networks (CNNs) to reduce the spatial dimensions (height and width) of the input tensor, while retaining the most important features. Pooling layers are often used after convolutional layers in CNNs, and they help to reduce the number of parameters in the network, which can prevent overfitting and reduce computational complexity. There are several types of pooling layers, including max pooling, average pooling, and global pooling. Max pooling is the most common type of pooling layer, which selects the maximum value from each subregion of the input tensor. Average pooling, on the other hand, computes the average value of each subregion of the input tensor. Global pooling computes a single value by taking the average or maximum of the entire feature max Pooling layers typically have two hyperparameters: the pooling size (the size of the subregion used for pooling) and the stride (the step size used to move the pooling window across the input tensor). A larger pooling size will result in greater spatial reduction, but may also result in loss of information, while a smaller pooling size may preserve more detail but lead to greater computational complexity. The stride parameter determines the amount of overlap between adjacent subregions

3.4 Flatten Layer

In convolutional neural networks (CNNs), a filtering layer, also known as a convolutional layer, is a type of layer that applies a set of filters to an input tensor. The filters are learned during the training process and are used to extract features from the input tensor. The filtering layer works by performing a convolution operation between the input tensor and the learned filters. This operation involves sliding the filters over the input tensor and computing the dot product between the filter and the portion of the input tensor it is currently covering. The result of the convolution operation is a feature map that represents the presence or absence of certain features in the input tensor. The filters used in a filtering layer can have different sizes and shapes, and the number of filters can also vary. The size and shape of the filters determine the spatial resolution of the output feature maps, while the number of filters determines the depth of the output feature maps. Filtering layers are often followed by activation layers, such as ReLU or sigmoid, to introduce nonlinearity into the network. They may also be followed by pooling layers to reduce the spatial dimensions of the output feature maps and control overfitting.

3.5 Dataset

In the first module, we developed the system to get the input dataset for the training and testing purpose. We have taken the dataset from Kaggle and soeme other out source. The dataset consists of 25296 **.**We will be using Python language for this. First we will import the necessary libraries such as Keras for building the main model, sklearn for splitting the training and test data. PIL for converting the images into array of numbers and other libraries such as pandas, numpy ,matplotlib and Tensorflow.

4. MATHEMATICAL MODEL

4.1 ALGORITHM 1: TF-IDF Estimation

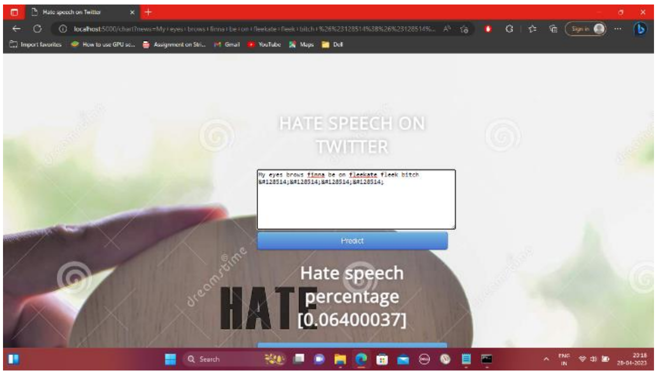
```
0: Start
1: Read the Preprocessed string
2: Divide string into words using space and store in a vector V
3: For i =0 to N (Where N is the length of V)
4: W= V[i]
5: Count W for the respective string as TF
6: Count W for the all other input strings that is DF
7: IDF= log (DF)
8: TF-IDF= TF* IDF
9: End For
10: Stop
```

4.2 ALGORITHM 2: Hidden Layer Estimation

```
//Input: Feature List FL, Weight set WS= { }
//Output: Hidden Layer value list HLV hiddenLayerEstimation
(FL, WS)
1: Start
2: HLV =∅ {Hidden Layer value]
3: for i=0 to size of FL
4: ROW= FL [i]
5: for j=0 to size of ROW
6: X=0 7: for k=0 to N [Number of Neurons]
8: ATR=ROW[j]
9: X = X + (ATR* WS[index])
10: index++
11: end for
12: HLV= reLUmax(0, X)
13: end for
14: end for
15: return HLV
16: Stop
```

5.RESULTS AND DISCUSSIONS

Our system will give the hate speech in percentage ratio. if this ratio is above the 0.5 percentage then our system consider the speech is hate speech. if this is ratio is less than 0.5 percentage then system detect has the normal speech.
Above the fig 3 show the speech has hate speech.
Because the ratio is below the 0.5 percentage.

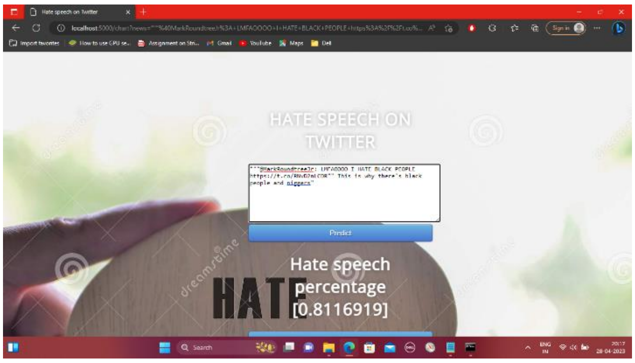


HATE SPEECH ON TWITTER

My eyes brows finna be on fleekate fleek bitch &#128514;&#128514;&#128514;&#128514;

Hate speech percentage [0.06400037]

Fig.3 Normal speech percentage



HATE SPEECH ON TWITTER

"""@MarkRoundtreeJr: LMFAOOOO I HATE BLACK PEOPLE https://t.co/RNvD2nLCOR"" This is why there's black people and niggers"

Hate speech percentage [0.8116919]

Fig.4 Hate speech percentage

In this paper, we focus on several famous deep learning methods. We then apply fusion methods to combine the classifiers to improve the overall classification performance.
Building a CNN architecture means that there are many hyperparameters to choose from, such as the input representations, number and sizes of convolution filters, pooling strategies, activation functions and so on. A few results highlight that max-pooling always beats average pooling and the ideal filter sizes are important but task-dependent. In this experiment several specific hyper parameters are set as suggestions.

6. CONCLUSIONS

Overall, the application is a pressing requirement, as the number of online social networks grows, as does the number of public shaming events, and as comments against site owners' callousness get louder. This research examines several applications of machine learning and hate speech identification to make shammers and shamming tweets easier to identify. After reviewing the literature, a new system can be presented that is capable of categorizing humiliating comments into distinct categories and blocking specific tweets if they are detected as humiliating.

REFERENCES

[1] M. Bojkovský and M. Pikuliak, ``STUFIIT at SemEval-2019 task 5: Multilingual hate speech detection on Twitter with MUSE and ELMo embeddings," in Proc. 13th Int. Workshop Semantic Eval., 2019, pp. 464468.

[2] M. S. Akhtar, A. Ekbal, and E. Cambria, ``How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [application notes]," IEEE Comput. Intell. Mag., vol. 15, no. 1, pp. 6475, Feb. 2020.

[3] O. de Gibert, N. Perez, A. García-Pablos, and M. Cuadros, ``Hate speech dataset from a white supremacy forum," in Proc. 2nd Workshop Abusive Lang. Online (ALW2), 2018, pp. 1120.

[4] T. Davidson, D. Bhattacharya, and I. Weber, ``Racial bias in hate speech and abusive language detection datasets," in Proc. 3rd Workshop Abusive Lang. Online, 2019, pp. 2535.

[5] T. Davidson, D. Warmsley, M. Macy, and I. Weber, ``Automated hate speech detection and the problem of offensive language," in Proc. ICWSM, May 2017, pp. 512515.

[6] E. Cambria, ``Affective computing and sentiment analysis," IEEE Intell. Syst., vol. 31, no. 2, pp. 102107, Mar./Apr. 2016.

[7] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, ``Deep learning for hate speech detection in tweets," in Proc. 26th Int. Conf. World Wide Web Companion WWW Companion, 2017, pp. 759760.

[8] H. Liu and L. Zhang, ``Advancing ensemble learning performance through data transformation and classiers fusion in granular computing context," Expert Syst. Appl., vol. 131, pp. 2029, Oct. 2019.

[9] H. Watanabe, M. Bouazizi, and T. Ohtsuki, ``Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," IEEE Access, vol. 6, pp. 1382513835, 2018.

BIOGRAPHIES

Pawase Shubhangi Dattatray, Fourth year BE(E&TC). Interested electronic sector and IT sector ,DL.

Pawar Gauri Vikas, Fourth year BE(E&TC). Interested electronic sector and IT sector, DL.

Sable Shweta Babasaheb, Fourth year BE(E&TC). Interested electronic sector and IT sector, DL.

Shinde Rajashree Nagnath, Fourth year BE(E&TC). Interested electronic sector and IT sector, DL.

Prof.dr.Jondhale Satish R.,M.E E&TC(VISL&Embedded System) phD Machine Learning & Specialization in Data Science