Journal of Computing, Science &Technology
https://focjournal.unidel.edu.ng/
editorjcst@unidel.edu.ng

ISSN: 3043-6818 Print
https://focjournal.unidel.edu.ng

A Deep Learning Framework for the Classification of Hate Speech in Tweets

Victor Eshiet Ekong[1], and Anietie Senam[2]
[1]Department of Software Engineering, Faculty of Computing University of Uyo, [2]Department of Computer Science, Faculty of Computing University of Uyo
victoreekong@uniuyo.edu.ng, senamanietie@uniuyo.edu.ng
Corresponding Author's Email *victoreekong@uniuyo.edu.ng*

Article Info
Date Received: 02-08-2024
Date Accepted: 12-10-2024

Keywords:
Auto encoder, Convolution Neural Network, Hate Speech, SoftMax, Regularizer

ABSTRACT

Hate speech is potentially harmful to individuals and the society. Social media remains a major channel for spreading hate speech. Posts on the social media are largely composed of non-standard linguistic signals, which makes automatic detection of hate speech on social media difficult. Poorly constructed linguistic contents on the social media contributes significantly to the difficulty of automatic detection of hate speech. Computational resources for creating large labeled corpora are costly. Deep neural network (DNN) presents an opportunity for efficiently learning features in a speech corpora therefore presenting prospect for automatic detection of hate speech. In this study an ensemble DNN model composed of a stacked auto encoder (SAE) and a convolutional neural network (CNN) is designed for the task of learning representations of X (formerly Twitter) comments with the aim of classifying hate speech. The dataset used in the study was obtained online from the X. The auto encoder (AE) component complements the weak feature extraction capability of CNN and improves data dimensionality reduction of the dataset. The output of the unsupervised AE and the extracted features, are input into the supervised CNN for classification. The study leveraged on the rich neural network support of Python to build and test the model through the low level libraries provided by Tensorflow and the high level neural network interface of Keras. The results showed that the ensemble AE-CNN had significant improvement for the binary classification task. The model achieved 96.0% accuracy and an F1-score of 94.8%.

1.0 INTRODUCTION

Hate speech is already a social malady world over. There is a constant struggle between freedom of speech and abuse of such freedom. With the proliferation of social media networks, hate speech has become much more easily propagated and spread .The anonymity and mobility offered by these social media platforms has made communication even more attractive given the ubiquitous and anonymous attributes it offers [1]. Cases of hate speeches are prevalent today due to social media proliferation and adoption by a large population as the best means of social dialogue. This is pertinent for countries where democracy is still nascent and more pronounced than developed democracies [2]. The development of a system to detect and screen hate speech in text messages can help in keeping the harmony in many countries. It is noteworthy that many extreme elements in the society only require smartphones and an internet connection to commit cyber hate posts [2]. Hate speech posts trend fast on the social media than any other post irrespective of geographic boundary [3].

A dynamic, automated and effective hate speech detection system, developed for social media networks becomes significantly important. More so, where a targeted individual or persons become vulnerable to harmful posts. Hate speech is known to be propagated via non-electronic and electronic media. The social media platforms facilitate information generation and circulation. The mobility and anonymity offered by it are strong reasons why people do not hesitate to pour out their feelings and invectives on perceived persons or groups resulting from hatred [4].

Advances in Artificial Intelligence (AI) have been deployed to provide an automatic means of detecting or recognizing comments and classifying them on social media platforms such as X (formerly known as Twitter) with the view to taking appropriate actions aimed at mitigating the effects of hate speech before they actually cause any societal harm [6]. These approaches exist in AI techniques such as natural language processing (NLP) and machine learning (ML). NLP, as a branch of AI provides routines for coding and decoding languages for computers and humans to interact using the natural language [7]. NLP utilize ML to automatically learn patterns in large language corpora and make a statistical inference. However, the limitations experienced in ML will also sufficiently affect NLP approach in classifying utterances on social media [6].

To select the deep features of the user datasets necessary for learning patterns in the data in order to achieve the desired classification, deep learning techniques are proposed [8]. Deep learning techniques have proven to be very powerful in classifying patterns in text data and the performance of the approaches outperforms classical, reinforcement and ensemble machine learning techniques [9].

In the remaining sections of this paper, Section 2 presents a

detailed review of related works. Section 3 presents the methodology adopted. In Section 4, we analyse and present results of experiments performed. Section 5 discusses findings from the results, and we conclude the research work in Section 6.

2.0 2. Related Works

Automatically detecting hate speech on social media networks is viewed as a text classification problem. Studies have shown that three methods are predominantly adopted; lexicon-based, classical learning and deep learning method.

2.1 Lexicon based methods

Lexical methods have been recognized for providing routines that identify offensive terminologies, however, they seldom identify hate speeches [10]. In [11], an LR was used with L2 regularisation to build a hate speech detection model that accurately distinguished between commonplace offensive language and serious hate speech. In [12], a two-step method was proposed for hate speech detection using paragraph2vec for joint modelling of comments and words. The distributed representations were learned with a CBOW natural language model and the embedding used to train a binary classifier that distinguished between hateful and non-hateful speech.
In [13], a lexicon based model was combined with a ML model to predict hate speech. The result provided a lexical baseline for the task of applying classification methods using annotated datasets.

2.2 Classical Learning Methods

Classical learning are predominant ML techniques that have deep rooted usage in classification tasks. In [14], a Gaussian NB, multinomial NB, RF and SVM is utilized to classify hate speech in a social media network. They tested and compared the performances of the different method combinations.
In [15], a multilingual hate speech classifier is developed. They utilized a GRU-based CNN, BERT, LR and mBERT on the following embedding; MUSE (Multi-Unit Spectral Expansion), translation and LASER.
In [16] a study is carried out to compare different supervised approaches to hate speech detection. They built models for binary classification subtasks recurrent RNNs, n-gram based NNs and a LSVC approach. For the Facebook task and the two cross-domain tasks the RNN model obtain a quite promising result, especially in the cross-domain setting. For X, they used an n-gram-based NN and the LSVC-based model. They adopted a supervised approach and performed grid search over different machine learning classifiers such as RNN, SVM and LR to select the best model for each task. Both n-gram-based and RNN models using embedding were tested. Results showed that while RNNs perform better in three of four tasks, classification on X data achieved a better ranking using the n-gram based RNN. With the emergence of word embedding techniques in DNN architectures for text classification tasks, several research studies have been carried out in its application in solving various problems including text classification.
In [17], n-grams, word-n-grams, word-skip-grams and a linear-SVM was utilized as classifiers to perform multi-class classification of hate speech in a SMN. Experimental results demonstrated that the main challenge lied in discriminating profanity and hate speech. In [18] an NN classifier called FastText is used to detect abusive text by employing SVMs with a linear kernel as their classification algorithm.

2.3 Deep Learning Based Methods

In [19] a survey was carried out on the different types of AEs that described various applications and use-cases of AEs to include; dimensionality reduction, recommendation systems, anomaly detection, clustering, classification, and as generative models.
In [20] a t-DeepHate is proposed, which connects the architecture with transfer learning methods that allows leveraging several smaller, unrelated data sets to train an embedding capable of representing “general purpose” hate speech. Their work was based on a DNN architecture called DeepHate capable of creating task-specific word and sentence embedding. This allowed for a higher performance in hate speech detection. DNN is proposed in [21] to efficiently estimate prediction uncertainty, thus filling the reliability of prediction gaps in text classification. Such that to reliably detect hate speech, they used LR and SVM from the scikit-learn library as the baseline classification models and LSTM network model as a baseline RNN with Monte Carlo dropout regularisation, which mimicked Bayesian inference within NN, and showed performance comparable to the best competing approaches using word embedding and superior performance using sentence embedding.

3.0 METHODOLOGY

The ML process flow described in [22, 4, 23] are studied and modified to suit our process flow design. The block diagram of the deep learning framework for hate speech detection is presented in Figure 1. The methodology of the framework begins with collection of data. The obtained tweet dataset motivated the next phase of data assessment and pre-processing. The model development followed suit resulting in the model being built and subjected to training and testing using the pre-processed dataset. The outcome from the model evaluated to None Hate Speech and Hate Speech.

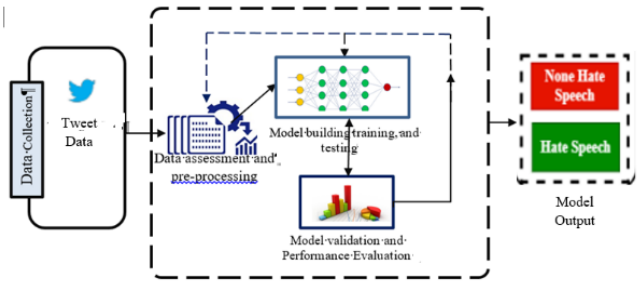

Figure 1: Block diagram of Deep Neural Network Hate Speech Classification

3.1 Data Collection

Tweet data created by [24, 25] was collected from X. The dataset had 13052 tweets. It is assessed and reduced to 16,914 annotated examples of two different classes labelled as either hate (Offensive/Hate speech consisting of Racist, Sexist and other forms of hate) or non-hate. Although the tweets were encoded with their respective identifiers (ids), only the contents of the dataset were extracted. The resulting 11,223 tweets accounted for 25 as racist; 2,990 as sexist and 8,208 containing no hate. In our methodology, we treat the hate speech detection as a binary classification. The reason being that, training a multi-input network is not straightforward as established by [25], hence we collapsed the sexist and racist classes into one hate class. We eventually had 3,015 hateful tweets and 8,208 non hateful tweets. Also to avoid oversampling in the experiment using dataset which understandably is made up of more sexist instances (2,990 out of 3,015), a handful of hateful tweets was added from another dataset from sexist instances (2,990 out of 3,015), a handful of hateful tweets was added from another dataset from the study in [25]. This balanced the datasets in the model; which would not learn to detect sexism type of hate at the expense of other forms of hate comments. So, any tweet that was 100% sexist was discarded and 300 extra samples of hate speech was added to the dataset.

This resulted in a final distribution of 3,315 hate (28.77%) and 8,208 non hateful (71.23%) tweets, with a total of 11,523 samples. Figures 2 (a) and (b) show the pie chart of the imbalanced and balanced dataset.

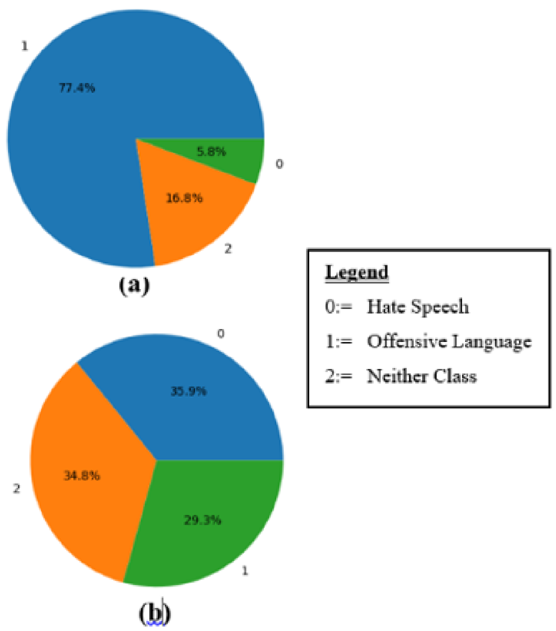

Figure 2: Plot showing the imbalanced (a) and balanced (b) datasets distribution

The remaining dataset was split into a training set and a test set in the ratio of 75:25.

Sample labelled X dataset is presented in Table 1.

Table 1: Labelled X Dataset for Hate Speech

No	Count	Hate Speech	Offensive Language		Class	Tweet
0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. & as a man you should always take the trash out...
1	3	0	3	0	1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
2	3	0	3	0	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit
3	3	0	2	1	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
4	6	0	6	0	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya 
5	3	1	2	0	1	!!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking with hoes! 😂😂😂"
6	3	0	3	0	1	!!!!!!"@__BrighterDays: I can not just sit up and HATE on another bitch.. I got too much shit going on!"
7	3	0	3	0	1	!!!!“@selfiequeenbri: cause I'm tired of you big bitches coming for us skinny girls!!”
8	3	0	3	0	1	" & you might not get ya bitch back & thats that "
...	...	...	...	...	...	...
24778	3	0	2	1	1	you's a muthaf***in lie “@LifeAsKing: @20_Pearls @corey_emanuel right! His TL is trash …. Now, mine? Bible scriptures and hymns”
24779	3	0	1	2	2	you've gone and broke the wrong heart baby, and drove me redneck crazy
24780	3	0	3	0	1	young buck wanna eat!!.. dat nigguh like I aint fuckin dis up again
24781	6	0	6	0	1	youu got wild bitches tellin you lies
24782	3	0	0	3	2	~~Ruffled | Ntac Eileen Dahlia - Beautiful color combination of pink, orange, yellow & white. A Coll http://t.co/H0dYEBvnZB
	80,383	6,052	40,819	13,612		

3.2 Deep Learning Model Design

We built a stacked (ensemble) auto-encoder convolution NN (AE-CNN) deep learning architecture as our model as shown Figure 3. The AE-CNN model is composed of; a word embedding layer connected to the AE followed by the CNN with three 1D convolution layer namely; convolutional1, convolutional2 and convolutional3; a batch normalization; an activation function; and a max pooling layer after which a SoftMax classifier layer is connected for classification with an activation function.

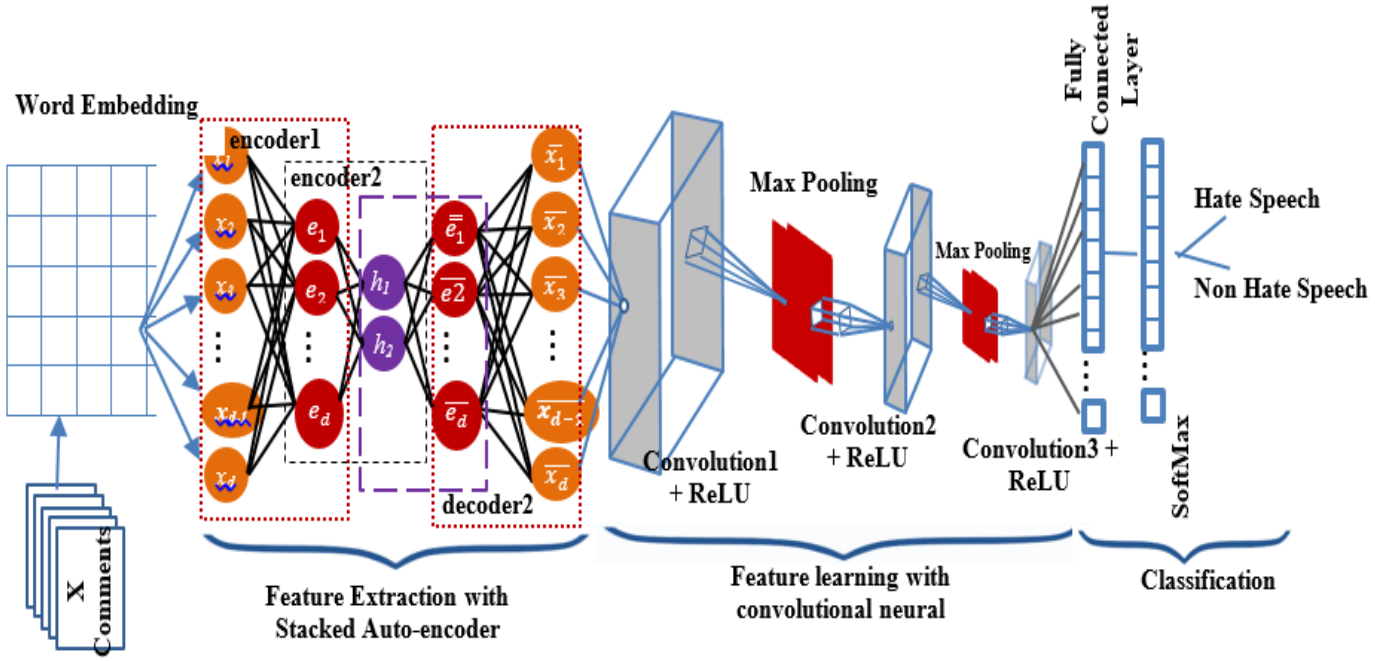

Figure 3: Stacked AE-CNN Deep Learning Architecture (Modified from

The first layer is a word embedding layer whose function is to map each tweet into a real vector domain. This is done by converting each word in a tweet to a real valued vector with fixed dimension with each element being the word's weight in that dimension. Each word sequence was limited to 100 words which was assumed to be long enough to accommodate tweets codes of any length, where tweets longer than expected were truncated and shorter tweets were zero-padded.

4.0 EXPERIMENT AND RESULTS

4.1 Training CNN for Feature Learning

CNNs are used to find general patterns in text and perform text classification. The output features obtained from the second AE is connected to the CNN layer to learn patterns from the features obtained from the dataset. The convolutional layer applies a moving window to input data and allows the model to learn the weights to apply to adjacent words thus learning about the correlation between nearby inputs. All operations were carried out on a single tweet at a time. A weighted average in a 10x10 window was applied to the embedded representation of the tweet, moving the window by 5 words, that is, STRIDE=5, and applying it again giving 4 such convolution results. The MaxPooling was then applied to the convolution results and had four results which were wired through a dense layer to the output layer. The sequences of word embeddings were passed through several convolutional operations, defined in the developed model as two convolutional layers with kernel heights of 3 and 4. These layers went through a ReLu activation and max-pooling operations. Finally, the max-values from the two different convolutional layers are concatenated and passed to a fully-connected layer and final to the SoftMax classification layer.

4.2 SoftMax Layer for Classification

The SoftMax algorithm is given in Equation 1.

$$\sigma(x)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad (1)$$

Where, σ = Softmax; x = input vector; e^{z_j} = standard exponential function for input vector; k = number of classes in the multi-class classifier and e^{z_k} = standard exponential function for output vector. SoftMax is the final layer of the model for the classification. It provided an accurate method to classify the tweet data.

The SoftMax layer of the model was trained to classify the 50-dimensional feature vectors into two classes namely, hate-class and non-hate-class. The AEs were trained in an unsupervised manner, the CNN and SoftMax layers were trained in a supervised fashion using labels for the training data. Four separate components of the stacked NN were trained in isolation (`autoenc1`, `autoenc2`, CNN and `softnet`). They are stacked together with the CNN and SoftMax layers to form a stacked network for classification. The results of the training were computed on the test set.

4.3 Plot of Training versus Validation Loss

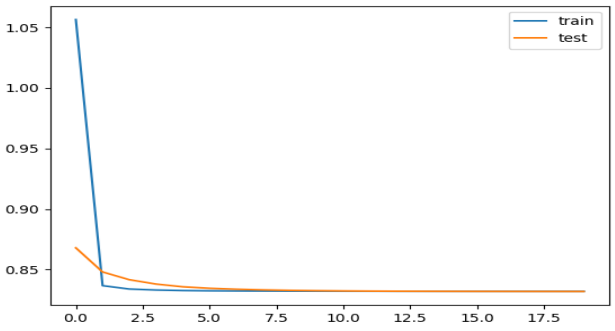

Figure 4: Plot of Training Versus Validation Loss

Figure 4 shows plot of the trained data in comparison with the test data (validation). A prediction on the trained model with the test data (25% of the dataset) consisting of about 2880 tweets was made and the model's data reconstruction was observed to be consistent.

Table 2: Performance metrics of the model for each fold

	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Fold-1	88.58	100.00	75.08	99.03
Fold-2	97.38	97.6	97.70	97.54
Fold-3	97.50	96.41	97.68	97.04
Fold-4	99.03	97.72	100.00	98.85
Average	96.01	97.88	92.62	94.80
[40]	88.4	100	77.1	87.1

Training and validation loss in Figure 4 are consistent and this means that the developed model is not over fitting and therefore, giving a good generalization capability of the model

4.4 Model Evaluation Metric

The performance of the AE-CNN classification model was analyzed to draw some conclusions. Also a comparative analysis was made between the AE-CNN model and similar state-of-the-Art model using the same dataset. The comparison was based on classification accuracy and F1 score. The model evaluation metric are shown in Equations 2, 3, 4 and 5.
Where t_p denotes the true positives, f_n denotes the false negatives, f_p denotes the false positives, and t_n denotes the true negatives. Accuracy is given in Equation 2:

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \quad (2)$$

Precision is given by Equation 3:

$$Precision(P) = \frac{t_p}{t_p + f_p} \quad (3)$$

Recall (or Sensitivity) is computed in Equation 4:

$$Recall(R) = \frac{t_p}{t_p + f_n} \quad (4)$$

F1-score, a standard measure of classification accuracy is given in Equation 5:

$$F1\ Score = \frac{2PR}{P+R} \quad (5)$$

4.5 Performance Analysis

The experimental results of the model on the test dataset are presented. Performance analysis of the model is conducted to compare the model's results with that of a similar model in [24]. The stacked AE-CNN classification model was trained four times, hence a 4-fold cross validation was used. Initially, the model was trained and tested with the dataset obtained by partitioning the original dataset into a ratio of 75:25 for training and test dataset. Table 2 shows the accuracy, precision, recall, and F1-score results for the classification.

The confusion matrix representing the model performance analysis is presented in Figure 5.

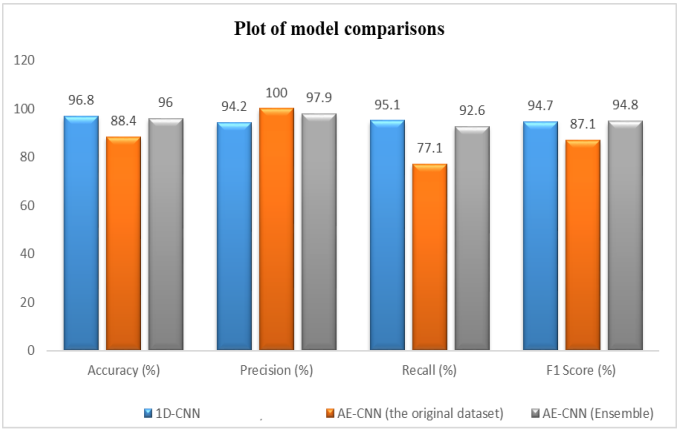

Figure 5: Confusion matrix showing model performance

Our model is further compared with a model from a similar study in [26]. They achieved an accuracy of 96.8% while our model had an accuracy of 96.0%. Table 3 and Figure 6 presents the summary of the model comparisons.

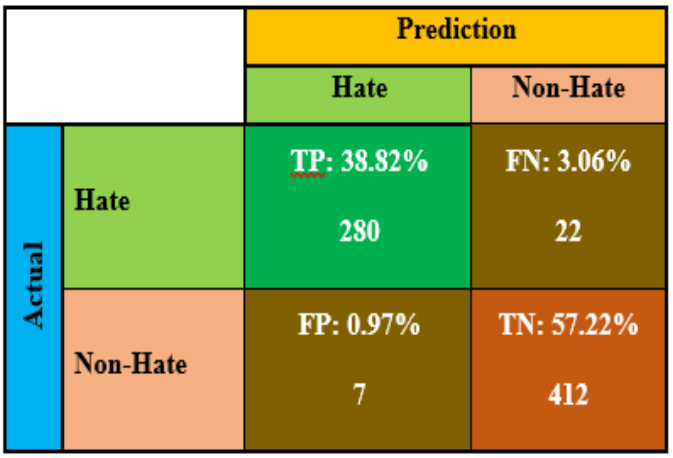

Table 3: Model comparison with similar classification models on X dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
1D-CNN-Global Vector classifier [26]	96.8	94.2	95.1	94.7
AE-CNN [24]	**88.4**	**100**	**77.1**	**87.1**
AE-CNN (Ensemble)	**96.0**	**97.9**	**92.6**	**94.8**

Figure 6: Model comparison

5.0 DISCUSSIONS

The proposed system shows good performance compared to that of similar studies in [26] and [24]. In [26] a one-dimensional CNN using 840 billion GloVE classifiers is proposed to classify hate speech in social media networks. They achieved an accuracy of 96.8% for this purpose using [24] original dataset baseline. Our AE-CNN ensemble model achieved almost the same accuracy and showed remarkable improvement with the baseline dataset. This is based mainly on the fact that our stacked AE ensemble classification model had the following strengths:

1. It provided the possibility to train each layer of the model separately, allowing for dimensionality reduction of X data features to be controllable.
2. The model extracts sample features useful for binary classification from the original dataset as input after two-layer AE training. It is worthy of note that if the original input data had some special features that were not related to the classification, it would have an impact on the final classification result.
3. The classification accuracy of the model is improved by the regularisation parameters added to each layer, which helps the CNN component of the model to achieve a good convergence by finding better local optima upon application of gradient descent.
4. The addition of a reconstruction loss as a regulariser in each layer also had an obvious impact on the regularisation effect. And finally. The stacked AE CNN is composed of multi-layer trained AEs, where each layer in the network is trained separately. And this is equivalent to initialising a reasonable value for the parameters of each layer in the network before the training which follows a waterfall approach. And by this, the network is easier to train and has faster convergence and higher accuracy.

6.0 CONCLUSION

This work focused on binary classification of hate speech using an ensemble AE-CNN model. The classification was performed on X dataset, classifying each tweet into either hate or non-hate class. An automated approach to feature extraction for text classification, targeting hate speech identification on the X social media was demonstrated. The ensemble AE-CNN model demonstrated an effective hate speech classification based on its accuracy rating of 96.0% and an F1-score of 94.8% as against the 88.4% and 87.7% respectively for an earlier study using the same original dataset. It also scaled close to a recent study in [26] that had an accuracy of 96.8%. We further conclude that better performance could be attained with appropriate fine-tuning of the model. A further study of could investigate other ML algorithms with tolerance for high dimensional datasets.

REFERENCES

[1] Zhang, Z.; Robinson, D.; and Tepper, J. (2018): *Detecting hate speech on Twitter using a convolution-GRU based deep neural network*. In The Semantic Web, pp. 745-760. Springer International Publishing, Cham.

[2] Bodrunova, S. S., A. Litvinenko, I. Blekanov, and D. Nepiyushchikh (2021), Constructive aggression? Multiple roles of aggressive content in political discourse on Russian youtube, *Media Commun*., 9(10):181–194.

[3] Slonje, R., P. K. Smith, and A. Frisén (2013) The nature of cyberbullying, and strategies for prevention, *Comput. Human Behav*., 29(1):26–32.

[4] Mullah, N. S and Zainon, W. M. Z (2021) Advances in Machine Learning Algorithms for Hate Speech detection in Social Media: A Review, *IEEE Access*, 20:1-14, DOI: 10.1109/ACCESS.2021.3089515.

[5] Fauzi, M. A., and Yuniarti, A. (2018). Ensemble method for Indonesian twitter hate speech detection. *Indonesian Journal of Electrical Engineering and Computer Science*, *11*(1): 294-299.

[6] Fortuna, P., Nunes, S. (2018): A survey on automatic detection of hate speech in text. *ACM Computing Surveys*, 51(4), DOI 10.1145/3232676

[7] Jurafsky, D., and Martin, J. H. (2020). *Speech and Language Processing.* 3rd ed. Pearson. USA.

[8] Goodfellow, I; Bengio, Y., and Courville, A. (2016). Deep Learning. *MIT Press.* ISBN: 978-0262035613.

[9] d'Sa, A. G., Illina, I. and Fohr, D. (2021)

Classification of Hate Speech Using Deep Neural Networks. *Revue d'Information Scientifique & Technique*, 2020, From Data and Information Processing to Knowledge Organization: Architectures, Models and Systems, 25 (01). hal-03101938.

[10] Gitari, N. D., Zuping, Z., Damien, H. and Long, J. (2015). A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215-230.

[11] Davidson, T., Warmsley, D., Macy, M. and Weber, I. (2017) Automated hate speech detection and the problem of offensive language. *arXiv* Preprint arXiv:1703.04009 (2017).

[12] Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., and Bhamidipati, N. (2015). Hate speech detection with comment embeddings. *Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion*, pp. 2-30.

[13] Martins, R., Gomes, M., Almeida, J. J., Novais, P. and Henriques, P. (2018). Hate speech classification in Social Media using emotional analysis, *In 7th Brazilian Conference on Intelligent Systems* (BRACIS).

[14] Laaksonen S. M., Haapoja, J., Kinnunen, T., Nelimarkka, M. and Pöyhtäri, R. (2020). The Datafication of Hate: Expectations and Challenges in Automated Hate Speech Monitoring. Front. *Big Data*, 3:3. doi: 10.3389/fdata.2020.00003

[15] Aluru, S. S., Mathew, B., Saha, P. and Mukherjee, A. (2020). Deep Learning models for multilingual hate speech detection. *arXiv* preprint arXiv:2004.06465.

[16] Corazza, M., Menini, S., Arslan, P., Sprugnoli, R., Cabrio, E., Tonelli, S. and Villata, S. (2018). Comparing Different Supervised Approaches to Hate Speech Detection, *EVALITA@CLiC-it*.

[17] Malmasi, S. and Zampieri, M. (2017). Detecting Hate Speech in Social Media, *Proceedings of Recent Advances in Natural Language Processing*, pp. 467–472, Varna, Bulgaria.

[18] Chen, H., McKeever, S. and Delany, S. (2017). Abusive text detection using neural networks. *AICS Conference*, Dublin, Ireland.

[19] Bank, D. Koenigstein, N. and Giryes, R. (2020). Auto-encoders. Available online at: https://www.researchgate.net/publication/339945889_Autoencoders

[20] Rizoiu M, Wang T, Ferraro G, Suominen H. (2019): Transfer Learning for Hate Speech Detection in Social Media. *CoRR*. Available online at: http://arxiv.org/abs/1906.03829 36

[21] Miok, K., Nguyen-Doan, D., Škrlj, B., Zaharie, D. and Robnik-Šikonja, M. (2019). Prediction Uncertainty Estimation for Hate Speech Classification. *Statistical Language and Speech Processing*, 11816, ISSN: 978-3-030-31371-5.

Emmanuel A. Dan, Bolanle F. Oladejo and Victor E. Ekong (2023) A Model For Predicting Food Insecurity in Nigeria using Deep Learning Technique, *Egyptian Computer Science Journal,* 47(1):1-10.

[23] Aljarah, I., Habib, M., & Castillo, P. A. (2020, February). Hate Speech Detection using Word Embedding and Deep Learning in the Arabic Language Context. In *ICPRAM*, pp. 453-460.

[24] Waseem, Z. and Hovy, D. (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.

[25] Founta, A-M., Chatzakou, D., Kourtellis, N., Blackburn, J., Vakali, A., and Leontiadis, I. (2018). Unified Deep Learning architecture for abuse detection. *In: Proceedings of the 10th ACM Conference on Web Science*, WebSci '19, pp. 105-114. ACM, New York, NY, USA.

[26] Das S, K. Bhattacharyya, and S. Sarkar (2023). Hate Speech Detection From Social Media Using One Dimensional CNN Coupled With Global Vector. *European Chemical Bulletin*, DOI: 10.48047/ecb/2023.12.si10.0081