Deep Learning based Hate Speech Detection on Twitter

Akshat Gaurav
Ronin Institute, Montclair, USA
akshat.gaurav@ieee.org

Brij B. Gupta
Department of Computer Science and Information Engineering
Asia University,
Taichung 413, Taiwan
bbgupta@asia.edu.tw

Kwok Tai Chui
Hong Kong Metropolitan University (HKMU),
Hong Kong
jktchui@hkmu.edu.hk

Varsha Arya
Department of Business Administration,
Asia University, Taiwan
111231027@live.asia.edu.tw

Priyanka Chaurasia,
University of Ulster, UK
p.chaurasia@ulster.ac.uk

2023 IEEE 13th International Conference on Consumer Electronics - Berlin (ICCE-Berlin) | 979-8-3503-2415-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCE-BERLIN58801.2023.10375620

***Abstract*—There have been growing worries about the effects of the widespread use of hate speech and harsh language on social media sites like Twitter. Effective strategies for recognising and reducing such dangerous material are necessary for resolving this problem. In this research, we give a detailed analysis of four deep learning models for identifying hate speech and inflammatory language on Twitter: the Long Short-Term Memory (LSTM), the Recurrent Neural Network (RNN), the Bidirectional LSTM (Bi-LSTM), and the Gated Recurrent Unit (GRU). We downloaded a large dataset from Kaggle that was curated for hate speech identification and used it in our experiment. We built each model after preprocessing and tokenization, then tweaked their hyperparameters for maximum efficiency. The models' abilities to detect hate speech were evaluated using standard measures including accuracy, precision, recall, and F1-score. Our findings show that there is a wide range of effectiveness amongst models in terms of identifying hate speech and inflammatory language on Twitter. In terms of accuracy and F1-scores, the Bi-LSTM and GRU models were superior to the LSTM and RNN. The results of this study imply that using bidirectional and gated processes may increase the models' capability of understanding the interdependencies and contexts of tweets, and hence, their classification accuracy.**

***Index Terms*—Hate Speech, LSTM, Bi-LSTM, GRU, RNN, Twitter**

I. Introduction

The rise of online social networking sites has greatly altered the way individuals all over the world interact socially and exchange information. Twitter, in particular, stands out as a potent tool for instantaneous information distribution and interaction across different platforms. Concerns about the negative effects of hate speech and abusive language on people, groups, and society at large have been raised in tandem with the meteoric expansion of online interactions [1], [2]. Developing reliable and effective strategies for identifying and limiting such harmful information on Twitter is crucial for resolving this urgent problem. The ever-evolving and ever-changing nature of internet communication makes it difficult to identify hate speech and inflammatory language [3]. Due to the sheer volume of new material being generated every day, manual moderation is just not feasible, making automatic detection approaches crucial. Due to their capacity to capture nuanced contextual connections in text data, deep learning models have emerged as viable techniques for natural language processing applications like hate speech identification [4], [5]. In recent years, researchers and data scientists have made significant strides in advancing deep learning techniques to tackle hate speech on social media platforms. The focus of this paper is to conduct a comparative analysis of four widely-used deep learning architectures: Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Bidirectional LSTM (Bi-LSTM), and Gated Recurrent Unit (GRU) models for hate speech and offensive language detection on Twitter.

II. Related Work

Many studies have been conducted, and many approaches have been proposed for identifying hate speech in online communities [6]. Finding that non-hateful, language-specific taboo interjections are misunderstood as indications of hate speech, this paper [7] employs a zero-shot, cross-lingual transfer learning framework for hate speech detection in English, Italian, and Spanish to identify hate speech against immigrants and women. Another study [8], proposes a CNN-based service framework called "HateClassify" for categorising social media content as hate speech, offensive, or neutral, and demonstrates that increasing hate speech detection by 20% is possible through the use of multilabel classification as opposed to multiclass classification. In this study [9], the author presents a comprehensive overview of existing machine learning (ML) algorithms and methodologies for hate speech identification in social media (SM), including traditional ML, ensemble approaches, and deep learning strategies. In another paper [10], author provides an ensemble-based semi-supervised learning methodology to enhance classification performance over supervised hate speech classification methods in a separate study, which makes use of the readily available social media information. Using a deep generative model, the authors of

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 26, 200)	3734800
lstm (LSTM)	(None, 64)	67840
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 128)	8320
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387

Total params: 3,811,347
Trainable params: 3,811,347
Non-trainable params: 0

Fig. 1. LSTM Model

a recent article [11] provided a dataset of 1 million hate and non-hate sequences, and then used this dataset to train a well-studied DL detector, achieving considerable performance increases on five different hate speech datasets.

III. Proposed Approach

A. Data Preprocessing

For the purposes of using deep learning models for the identification of hate speech and inflammatory language on Twitter, data preprocessing plays an essential role in preparing the raw text data for analysis. The goal of the data preparation procedures is to standardise the text and clean it up by getting rid of any extraneous information or noise.

- The first step in the preprocessing pipeline is the removal of HTML entities and Unicode emojis from the text. These elements do not contribute to the semantic meaning of the tweets and can introduce unnecessary noise.
- User tags (e.g., "@username") are replaced with a generic term "user" to eliminate user-specific information while preserving the overall tweet content.
- URLs are removed from the text as they do not add meaningful information and can distract the models from focusing on the tweet's content.
- Unnecessary symbols, such as double quotes, single quotes, exclamation marks, and backticks, are also removed to ensure that only relevant words and context remain.
- Common stopwords, such as articles and conjunctions, are filtered out from the text.

B. LSTM Model

The LSTM model architecture presented in Figure 1 is designed for hate speech and offensive language detection on Twitter. The first layer in the model is the Embedding layer, which is responsible for mapping the words in the input text to their corresponding dense word vectors. This step allows the model to represent words in a continuous vector space, capturing the semantic relationships between words. Following the LSTM layer, a Dropout layer with a rate of 0.5 is applied to further regularize the model and prevent overfitting. This additional dropout helps in generalizing the model's learned features to new, unseen data, enhancing its performance on different hate speech detection tasks. A Dense layer with 128 neurons and a ReLU activation function is added after the Dropout layer. This dense layer allows the previous output from the LSTM to be connected and transformed further, enabling the model to capture complex relationships between different features and refine its representation of the input text. Another Dropout layer with a rate of 0.5 is employed before the final output layer. This dropout layer serves the same purpose as the previous ones, promoting better generalization and robustness of the model.

The output layer consists of three neurons with a softmax activation function. The softmax function converts the model's raw outputs into probabilities, representing the likelihood of each class (0, 1, 2), which correspond to the different categories of hate speech: non-hate speech, offensive language, and hate speech.

C. Bi-LSTM Model

The Bi-LSTM model architecture presented in Figure 2 is another approach for hate speech and offensive language detection on Twitter, utilizing Bidirectional Long Short-Term Memory (Bi-LSTM) layers. Bi-LSTM layers enhance the traditional LSTM model by considering both forward and backward information, allowing the model to better capture the context and dependencies within the input text. Similar to the previous LSTM model, the Bi-LSTM model begins with an Embedding layer, which maps words in the input text to dense word vectors, enabling the model to understand the semantic relationships between words effectively.

The Bidirectional layer wraps the LSTM layer, effectively creating two separate LSTM layers for processing the input text in both forward and backward directions. This bidirectional processing enables the model to learn from the sequential context in both past and future words, which enhances its ability to understand the full context of a given word within the sentence. Similar to the LSTM model, the Bi-LSTM model is compiled using the Adam optimizer, categorical cross-entropy loss function, and various metrics (accuracy, F1-score, precision, and recall) for evaluation during training and testing.

D. GRU Model

The GRU (Gated Recurrent Unit) model architecture presented in Figure 3 is a variant of the LSTM model. Like LSTM, GRU is another type of recurrent neural network that can effectively process sequential data while mitigating some of the computational complexity. The GRU model starts with an Embedding layer, similar to the previous models, which maps the words in the input text to dense word vectors, capturing semantic relationships between words. The GRU layer, consisting of 64 GRU cells, is the core component of this architecture. The GRU units are equipped with gating mechanisms that allow them to control the flow of information

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 26, 200)	3734800
bidirectional_1 (Bidirectional)	(None, 128)	135680
dropout_4 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 128)	16512
dropout_5 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 3)	387

Total params: 3,887,379
Trainable params: 3,887,379
Non-trainable params: 0

Fig. 2. Bi-LSTM Model

Layer (type)	Output Shape	Param #
embedding_5 (Embedding)	(None, 26, 200)	3734800
gru_1 (GRU)	(None, 64)	51072
dropout_8 (Dropout)	(None, 64)	0
dense_8 (Dense)	(None, 128)	8320
dropout_9 (Dropout)	(None, 128)	0
dense_9 (Dense)	(None, 3)	387

Total params: 3,794,579
Trainable params: 3,794,579
Non-trainable params: 0

Fig. 3. GRU Model

in and out of the memory cells, effectively managing the context and dependencies within the input text. The dropout rate of 0.3 is applied during training to prevent overfitting, ensuring that the model does not memorize specific examples and can generalize well to new data.

E. Simple RNN Model

The RNN (Recurrent Neural Network) model architecture presented in Figure 4 is a basic recurrent model designed for hate speech and offensive language detection on Twitter. RNNs are particularly suited for sequential data processing, as they can maintain a hidden state that allows them to capture the temporal dependencies within the input text. The SimpleRNN layer is the core component of this architecture. It consists of 64 SimpleRNN cells, each of which maintains a hidden state that stores information from the previous time step. This allows the model to consider the sequential context of the text during training and inference. To prevent overfitting, a dropout rate of 0.3 is applied during training, which randomly deactivates neurons, encouraging the model to learn more robust representations. Following the SimpleRNN layer, a Dropout layer with a rate of 0.5 is added to further prevent overfitting. This dropout layer aids in improving generalization

Layer (type)	Output Shape	Param #
embedding_8 (Embedding)	(None, 26, 200)	3734800
simple_rnn_2 (SimpleRNN)	(None, 64)	16960
dropout_14 (Dropout)	(None, 64)	0
dense_14 (Dense)	(None, 128)	8320
dropout_15 (Dropout)	(None, 128)	0
dense_15 (Dense)	(None, 3)	387

Total params: 3,760,467
Trainable params: 3,760,467
Non-trainable params: 0

Fig. 4. RNN Model

and prevents the model from becoming too reliant on specific examples.

IV. Results and Disscussion

The experiment was conducted on a Kaggle notebook, utilizing the computational resources and software environment provided by Kaggle. The models, including LSTM, RNN, Bi-LSTM, and GRU, were built using the Keras library with a TensorFlow backend. Python served as the programming language, offering a rich ecosystem of machine learning libraries and tools. The dataset for hate speech and offensive language detection on Twitter was sourced from Kaggle. Data preprocessing, model implementation, and evaluation were performed within the Kaggle notebook, which provided a seamless and well-configured environment for conducting the experiment.

The dataset used in this study for hate speech and offensive language detection on Twitter was classified into three categories: hate speech, offensive language, and neither (non-hate speech). Figure 5 representation of the dataset shows the distribution of instances across these three classes.

A. Confusission Matrix

The confusion matrices for each of the models, namely LSTM, Bi-LSTM, GRU, and RNN, display the performance of hate speech and offensive language detection on Twitter across different categories: hate speech, offensive language, and non-hate speech. Starting with the LSTM model represented in **??**, demonstrated relatively balanced performance with moderate accuracy in predicting hate speech (67 instances correctly classified), offensive language (3547 instances correctly classified), and non-hate speech (603 instances correctly classified). However, the model showed higher misclassifications in hate speech and non-hate speech categories, with some instances being wrongly classified as offensive language.

The Bi-LSTM model in Figure 7, appeared to struggle significantly in correctly classifying instances across all categories. It seemed to heavily favor the offensive language class, as evidenced by a large number of instances being

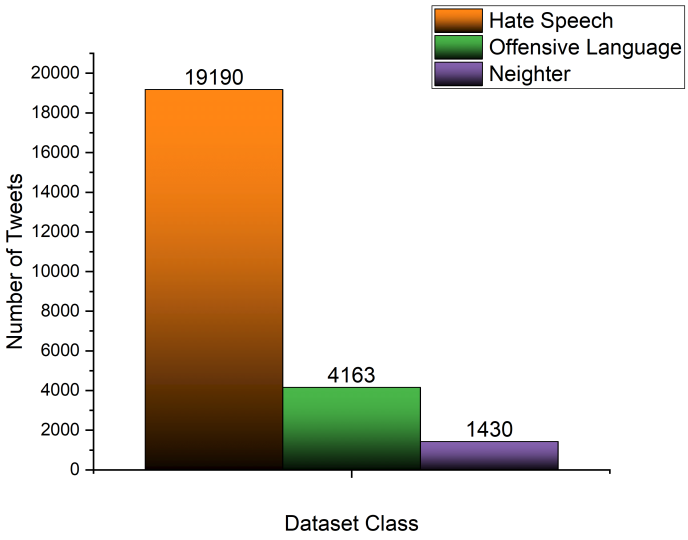

Fig. 5. Dataset Classes

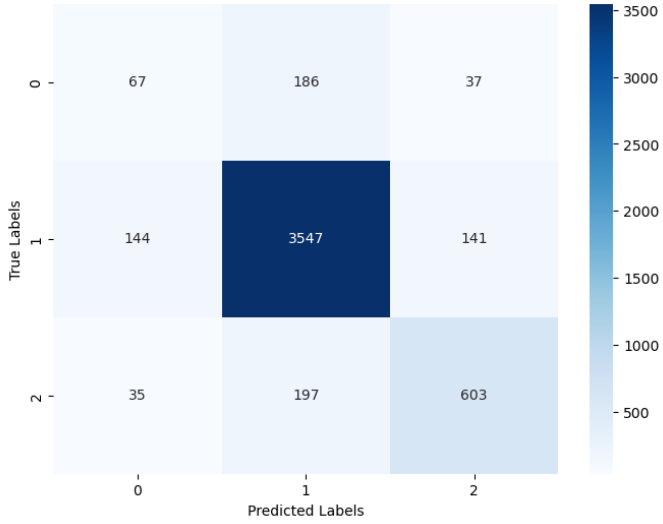

Fig. 6. LSTM Model

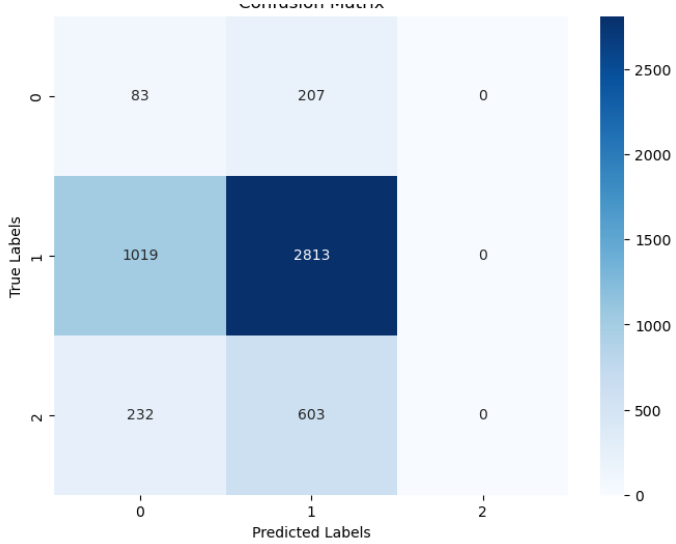

Fig. 7. Bi-LSTM Model

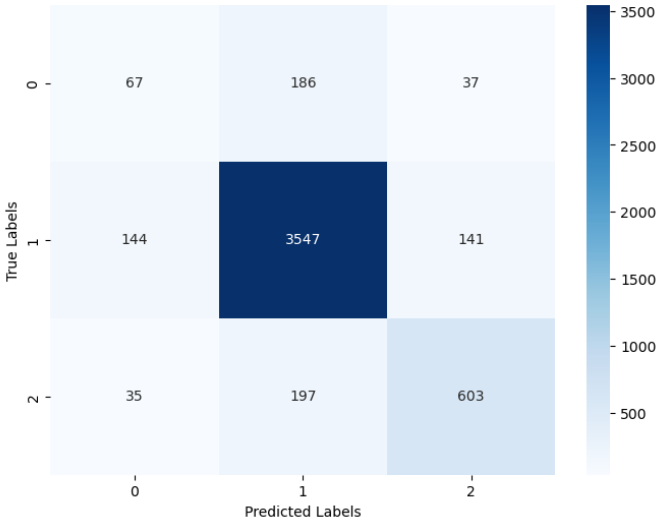

Fig. 8. GRU Model

misclassified as offensive language. Additionally, it completely failed to predict instances in the neither (non-hate speech) category.

In comparison, the GRU model in Figure 8 achieved similar performance to the LSTM model, albeit with slightly fewer correct classifications. While it was able to correctly classify instances across all categories, it also encountered challenges in hate speech and non-hate speech classifications, similar to the LSTM model.

Lastly, the RNN model in Figure 9 displayed a balanced performance, achieving moderate accuracy in hate speech (69 instances correctly classified), offensive language (3480 instances correctly classified), and non-hate speech (631 instances correctly classified). However, like the LSTM and GRU models, it faced difficulties in correctly predicting instances in the hate speech and non-hate speech categories.

B. Performence Matrices

The performance metrics for each model, including precision, recall, and F1-score, provide valuable insights into their effectiveness in hate speech and offensive language detection on Twitter.

The LSTM model achieved an overall accuracy of 85% as represented in Figure 10, demonstrating good performance in correctly classifying instances across all categories. It showed the highest precision and recall values for the offensive language class (class 1), indicating that the model correctly identified offensive language instances with high confidence. However, its performance was comparatively lower for the hate speech (class 0) and non-hate speech (class 2) categories.

However, the Bi-LSTM model exhibited lower overall accuracy of 58% as represented in Figure 11, primarily due to its

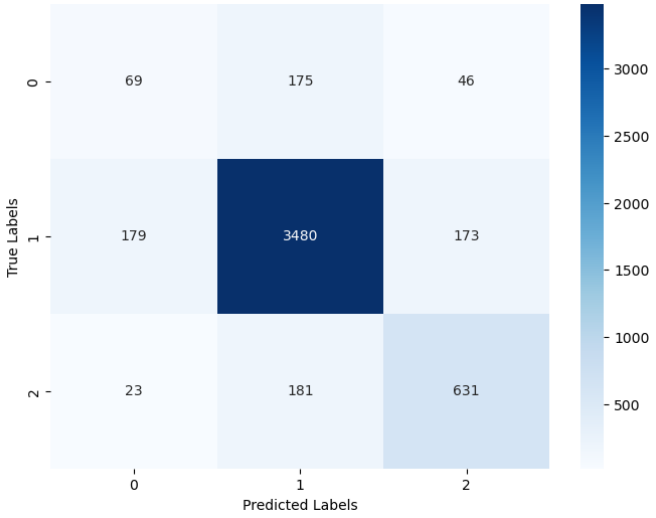

Fig. 9. RNN Model

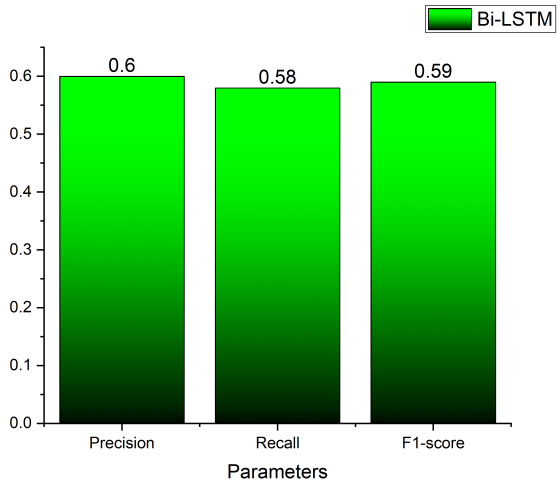

Fig. 11. Bi-LSTM Model

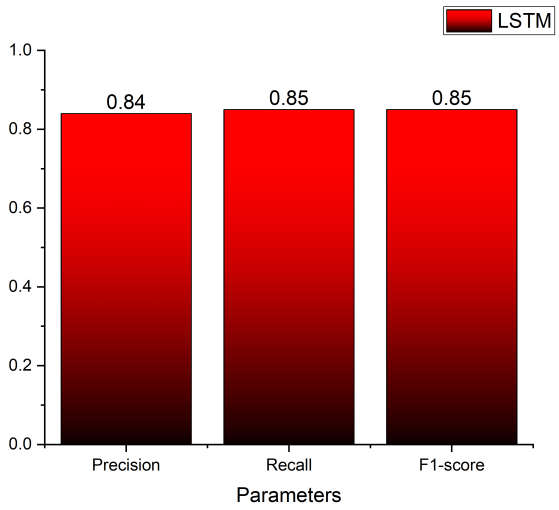

Fig. 10. LSTM Model

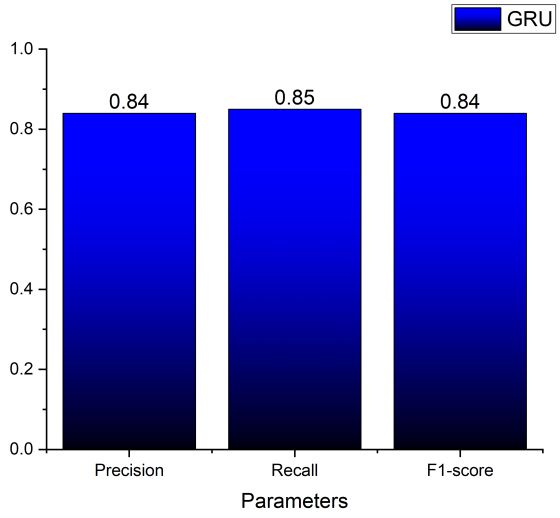

Fig. 12. GRU Model

significant struggles in correctly classifying instances in the neither (non-hate speech) category (class 2). The precision, recall, and F1-score for class 2 were extremely low, indicating that the model failed to correctly predict instances belonging to the non-hate speech class.

Similarly, the GRU model achieved an accuracy of 85% (fig:enter-grulstm-f) and showed balanced performance in identifying hate speech (class 0) and offensive language (class 1) instances. However, its precision and recall values for the non-hate speech (class 2) category were comparatively lower, suggesting some difficulty in correctly classifying instances in this category. The RNN model in Figure 13 also achieved an accuracy of 84% and demonstrated balanced performance for hate speech (class 0) and offensive language (class 1) instances. It performed well in predicting the non-hate speech (class 2) category, with relatively higher precision and recall values.

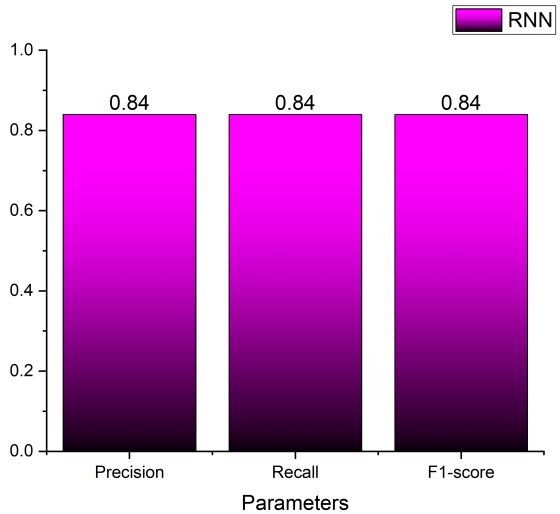

Fig. 13. RNN Model

V. Conclusion

This paper presented a comprehensive investigation into the detection of hate speech and offensive language on Twitter, employing four deep learning models: LSTM, Bi-LSTM, GRU, and RNN. The study utilized a dataset with hate speech, offensive language, and non-hate speech instances collected from Kaggle. Upon evaluating the models, we observed varying levels of effectiveness in hate speech detection. The LSTM and GRU models demonstrated balanced performance, achieving higher accuracy and effectively identifying offensive language instances. The RNN model also showed competitive results, displaying balanced accuracy across all categories. However, the Bi-LSTM model faced significant challenges, particularly in correctly classifying non-hate speech instances, leading to lower overall accuracy. This highlights the limitations of the Bi-LSTM architecture in handling imbalanced class distributions.

References

[1] K. Pareek, A. Choudhary, A. Tripathi, K. Mishra, and N. Mittal, "Hate and aggression detection in social media over hindi english language," *International Journal of Software Science and Computational Intelligence (IJSSCI)*, vol. 14, no. 1, pp. 1–20, 2022.

[2] S. Joshi, S. Koparrati, and V. Singhal, "Marketing analytics for analyzing social media and branding efforts of indian telecom operators," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 12, no. 1, pp. 1–18, 2022.

[3] J. Gu, N. D. Vo, J. J. Jung *et al.*, "Contextual word2vec model for understanding chinese out of vocabularies on online social media," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 18, no. 1, pp. 1–14, 2022.

[4] Z. Zhang, R. Sun, C. Zhao, J. Wang, C. K. Chang, and B. B. Gupta, "Cyvod: a novel trinity multimedia social network scheme," *Multimedia Tools and Applications*, vol. 76, pp. 18 513–18 529, 2017.

[5] N. Kumar, V. Poonia, B. Gupta, and M. K. Goyal, "A novel framework for risk assessment and resilience of critical infrastructure towards climate change," *Technological Forecasting and Social Change*, vol. 165, p. 120532, 2021.

[6] S. K. Mohapatra, S. Prasad, D. K. Bebarta, T. K. Das, K. Srinivasan, and Y.-C. Hu, "Automatic hate speech detection in english-odia code mixed social media data using machine learning techniques," *Applied Sciences*, vol. 11, no. 18, p. 8575, 2021.

[7] D. Nozza, "Exposing the limits of zero-shot cross-lingual hate speech detection," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021, pp. 907–914.

[8] M. U. Khan, A. Abbas, A. Rehman, and R. Nawaz, "Hateclassify: A service framework for hate speech identification on social media," *IEEE Internet Computing*, vol. 25, no. 1, pp. 40–49, 2020.

[9] N. S. Mullah and W. M. N. W. Zainon, "Advances in machine learning algorithms for hate speech detection in social media: a review," *IEEE Access*, vol. 9, pp. 88 364–88 376, 2021.

[10] S. Alsafari and S. Sadaoui, "Ensemble-based semi-supervised learning for hate speech detection," in *The International FLAIRS Conference Proceedings*, vol. 34, 2021.

[11] T. Wullach, A. Adler, and E. Minkov, "Towards hate speech detection at large via deep generative modeling," *IEEE Internet Computing*, vol. 25, no. 2, pp. 48–57, 2020.