International Journal of Engineering, Science & InformationTechnology (IJESTY)
Volume 4, No. 1 (2024) pp. 39-43
ISSN 2775-2674 (online)
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v4i1.491
Research Paper, Short Communication, Review, Technical Paper

Comparative Analysis of CNN-RNN Models for Hatespeech Detection Incorporating L2 Regularization

Tri Pratiwi Handayani*, Wahyudin Hasyim

Department of Information System, Faculty of Science and Computer, Universitas Muhammadiyah Gorontalo, Gorontalo, Indonesia

**Corresponding author Email: tripratiwi@umgo.ac.id*

The manuscript was received on 11 August 2023, revised on 27 December 2023, and accepted on 10 January 2024, date of publication 20 January 2024

Abstract

This study aims to address the challenge of detecting hate speech in text data by comparing two experimental CNN-RNN models. The primary issue is achieving a balance between precision and recall in hate speech detection while preventing overfitting and ensuring good generalization. Two different approaches were applied: the first model used standard training techniques, while the second model incorporated L2 regularization and early stopping. The research involved using Keras Tokenizer for text tokenization, layering with CNN and LSTM for feature extraction and temporal context capturing, and applying dropout to prevent overfitting. L2 regularization and early stopping were added to the second model to enhance generalization. The findings reveal that the first model, although exhibiting some overfitting, attained a higher overall accuracy of 78% and more balanced F1-scores for both the "Not Hate Speech" and "Hate Speech" categories. The second model, although achieving higher precision for hate speech (0.81), had lower recall (0.58), resulting in an overall accuracy of 75%. This suggests that regularization and early stopping need careful tuning to avoid reducing sensitivity to hate speech detection.

Keywords: *CNN-RNN, Hate Speech, Regularization, LSTM.*

1. Introduction

The proportion of students who have reported being victims of cyberbullying has surged by 55% since 2015 and has more than tripled since 2007. About 45.5% of middle and high school students have experienced cyberbullying at least once in their live [1]. This alarming trend necessitates effective detection mechanisms for hate speech, which remains a challenging task due to the need to balance precision and recall while preventing overfitting. To tackle these challenges, this study focuses on leveraging machine learning, particularly deep learning techniques, to enhance hate speech detection. By comparing two experimental hybrid CNN-RNN models—one using standard training techniques and the other incorporating L2 regularization and early stopping—the study aims to identify the most effective strategy for improving model performance and generalization. The objective is to evaluate the effectiveness of CNN-RNN models with standard training techniques versus those with L2 regularization and early stopping in detecting hate speech. The goal is to determine which approach achieves better balance in precision and recall, and how they handle overfitting to enhance generalization. Convolutional Neural Networks (CNNs) are proficient at capturing spatial features and recognizing local patterns, such as phrases and key words, making them ideal for text processing tasks like sentiment analysis and hate speech detection [2], [3]. Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, are highly effective at managing sequential data by retaining memory of previous inputs, which helps in understanding context and dependencies [4]–[6]. Combining CNNs and RNNs leverages their strengths in feature extraction and sequence modeling. However, issues such as class imbalance and overfitting persist [7], [8]. L2 regularization aids by penalizing large weights, promoting simpler models that generalize more effectively. Early stopping prevents overfitting by ceasing training once the validation performance ceases to improve [4], [9], [10]. The first model is anticipated to serve as a baseline, highlighting the strengths and weaknesses of the standard approach. In contrast, the second model is expected to demonstrate the advantages of regularization and early stopping in achieving a better balance between precision and recall.The insights from this research are anticipated to contribute to the development of more robust and accurate hate speech detection models, ultimately enhancing efforts to combat online cyberbullying effectively.

2. Literature Review

Hate speech detection on social media is a pressing issue in Indonesia, with significant advancements being made using machine learning techniques such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Recent studies have demonstrated the effectiveness of these methods in classifying hate speech in Indonesian language tweets and Instagram comments. For instance, a study utilizing CNN, RNN, and hybrid RNN-CNN models with GloVe feature expansion found the RNN model [11], while another study employing a modified TextCNN with word2vec skip-gram models for detecting hate speech in Instagram comments [12], [13]. Comparative research on BiLSTM, CNN, and RNN architectures also highlighted the superior performance of RNN with BERT embeddings [14]. Moreover, a CNN model was found to outperform LSTM and CNN+LSTM combinations [15].
Despite these advancements, there remains a research gap in achieving a balance between precision and recall in hate speech detection while preventing overfitting and ensuring good generalization. This study aims to address this challenge by comparing two experimental CNN-RNN models. The primary issue is balancing precision and recall in hate speech detection. Two different approaches were applied, the first model used standard training techniques, while the second model incorporated L2 regularization and early stopping. The research involved using Keras Tokenizer for text tokenization, layering with CNN and LSTM for feature extraction and temporal context capturing, and applying dropout to prevent overfitting. L2 regularization and early stopping were added to the second model to enhance generalization

3. Method

3.1. Data Collection

The datasets used were obtained from [11], consisting of new_kamusalay.csv, abusive.csv, and data.csv. The primary dataset, data.csv, was sampled to a size of 1000 instances for both experiments. This smaller sample size was chosen to ensure the experiments could be conducted efficiently within the constraints of available computational resources and tim. Using a more manageable dataset size allowed for rapid iteration and testing of the models, facilitating a more focused evaluation of the different training techniques and model architectures without sacrificing the integrity of the results [16]. Additionally, this approach helps to streamline the preprocessing and training processes, making it feasible to apply advanced techniques such as L2 regularization and early stopping within a reasonable timeframe [17].

3.2. Data Cleaning

The cleaned text was processed using Keras's Tokenizer [18], which converts the text into sequences of integers, with each integer representing a specific word based on its frequency in the dataset. This tokenization process builds a comprehensive dictionary of words. After tokenization, the sequences were padded to ensure uniform length across all inputs. Padding involves appending zeros to the end of shorter sequences to match the length of the longest sequence. This is essential for neural networks, as they require inputs of uniform size for optimal performance [19]–[21].
The dataset labels, indicating whether a tweet is hate speech or not, were one-hot encoded. This method transforms the categorical labels into a binary matrix representation, where each label is represented by a vector with a single element set to "1" (indicating the class) and all other elements set to "0.". This format is particularly suitable for training neural networks, enabling them to efficiently process and classify the data.

3.3. Data Train-Test Split

The dataset was subsequently divided into training and testing sets in an 80-20 ratio to ensure the model could be effectively trained and evaluated on separate data, thereby improving its ability to generalize to new data [22]

3.4. Class Weights Calculation

To tackle the problem of class imbalance in the dataset, class weights were calculated and incorporated during the training process [23] .This computation involves determining the relative frequencies of each class and assigning a weight to each class that is inversely proportional to its frequency. By doing this, the model is encouraged to pay more attention to the minority class, ensuring that instances of hate speech, which may be less frequent than non-hate speech, are adequately represented during training.

3.5. Oversampling

To further address class imbalance, oversampling was applied using the Random Over Sampler technique. This method involves balancing the classes by increasing the number of instances in the minority class through the random duplication of existing instances [24]. By doing so, the model is exposed to an equal number of examples from both classes during training, which helps it learn to identify hate speech as effectively as non-hate speech. This approach ensures that the model does not become biased towards the majority class and improves its ability to generalize and perform well on imbalanced data [25].

3.6. Model Architectures and Training

3.6.1. First Model

The architecture of the first model was designed to combine the strengths of CNNs and RNNs for effective text classification. It begins with an embedding layer that transforms words into dense vectors, capturing their semantic meanings and relationships [26]. This is followed by a Conv1D layer that extracts features from the input sequences through convolutional filters, identifying significant local patterns within the text [27]. The MaxPooling1D layer subsequently reduces the dimensionality of the feature maps, retaining the most crucial features while minimizing computational complexity [28]. An LSTM layer follows, capturing temporal dependencies to understand the context and order of words, essential for accurate classification [29]. A dropout layer is included to mitigate overfitting by randomly dropping neurons during training, enhancing the model's generalization to new data [30]. Finally, a dense layer with a softmax activation function produces the output, suitable for distinguishing between hate speech and non-hate speech [31]. The model was trained for 10 epochs with class weights applied to address class imbalance, ensuring improved detection capabilities for hate speech by giving adequate attention to the minority class [32].

3.6.2. Second Model

The second model id Similar to the first model, it includes an Embedding layer for converting words into dense vectors, a Conv1D layer for extracting features from the input sequences using convolutional filters, a MaxPooling1D layer for reducing the dimensionality of the feature maps, and an LSTM layer for capturing temporal dependencies in the text. Additionally, it includes a Dropout layer to reduce overfitting by randomly deactivating neurons during training. The key difference is the inclusion of L2 regularization in the Dense layer [17]. This regularization technique penalizes large weights, encouraging the model to maintain smaller weights, thereby reducing the risk of overfitting by preventing the model from becoming overly complex and fitting the noise in the training data. The training process was designed to run for up to 20 epochs, with an early stopping mechanism in place [32]. The second model is trained for 20 epochs compared to 10 epochs for the first model due to several factors. The second model incorporates L2 regularization in the dense layer, which helps prevent overfitting by penalizing large weights, allowing for a longer training period. Additionally, the early stopping mechanism ensures the model halts training once the validation loss stops improving, balancing sufficient training and preventing overfitting. Early stopping monitors the validation loss during training and halts the process if the validation loss does not improve for a specified number of epochs. This ensures the model stops training at the optimal point where it best generalizes to the validation set, thereby avoiding overfitting. Class weights were also applied to handle class imbalance, ensuring the model pays adequate attention to the minority class, improving its detection capabilities for hate speech.

4. Result and Discussion

4.1. Result

The performance of the first model was evaluated using several metrics, including precision, recall, F1-score, and Support. The results is described in Table 1

Table 1. Evaluation result of the first model

Class	Precision	Recall	F1-score	Support
Not Hate Speech	0.77	0.84	0.80	110
Hate Speech	0.78	0.70	0.74	90
Accuracy			0.78	200
Macro avg	0.78	0.77	0.77	200
Weighted avg	0.78	0.78	0.77	200

The model demonstrated a high recall of 0.84 for the "Not Hate Speech" class, meaning it correctly identified 84% of the actual non-hate speech instances. The precision of 0.77 indicates that 77% of the instances the model predicted as non-hate speech were correct. The F1-score, which is the harmonic mean of precision and recall, was 0.80, indicating a balanced performance between precision and recall for this class. For the "Hate Speech" class, the model achieved a precision of 0.78, meaning 78% of the instances predicted as hate speech were correct. The recall was 0.70, indicating that the model correctly identified 70% of the actual hate speech instances. The F1-score was 0.74, reflecting a slightly lower but still balanced performance compared to the "Not Hate Speech" class. The model achieved an overall accuracy of 0.78, meaning that it correctly classified 78% of the instances in the test set, which included a total of 200 instances. The macro average is the unweighted mean of the precision, recall, and F1-score for both classes. It offers an overall performance measure without accounting for class imbalance. The precision, recall, and F1-score values, all around 0.77, indicate consistent performance across both classes. The weighted average considers the support (number of instances) for each class, offering a more balanced performance measure that reflects the class distribution. The precision and recall are both 0.78, with the F1-score slightly lower at 0.77, indicating that the model's performance is robust and consistent when accounting for the prevalence of each class in the dataset. Overall, the first model demonstrated strong performance in detecting both hate speech and non-hate speech, with a balanced precision, recall, and F1-score for both classes. The higher recall for non-hate speech suggests the model is particularly effective at identifying non-hate speech instances, while the precision and F1-score indicate a good balance in performance across the board.
The performance of the second model was evaluated using several metrics, including precision, recall, F1-score, and Support. The results is described in Table 2

Table 2. Evaluation result of the Second model

Class	**Precision**	**Recall**	**F1-score**	**Support**
Not Hate Speech	0.72	0.89	0.80	110
Hate Speech	0.81	0.58	0.68	90
Accuracy			0.75	200
Macro avg	0.77	0.73	0.74	200
Weighted avg	0.76	0.75	0.74	200

Table 2 outlines the evaluation metrics for a text classification model aimed at detecting hate speech. These metrics include precision, recall, and F1-score for each class (Not Hate Speech and Hate Speech), along with their respective support values. Additionally, it provides overall accuracy, macro average, and weighted average metrics. For the Not Hate Speech class, the precision is 0.72, indicating that 72% of instances predicted as Not Hate Speech are accurate. The recall is 0.89, showing that the model correctly identifies 89% of all actual Not Hate Speech instances. The F1-score, which balances precision and recall, is 0.80, demonstrating a good balance between the two. The support value, representing the number of actual instances in the dataset, is 110. For the Hate Speech class, the precision is higher at 0.81, meaning 81% of instances predicted as Hate Speech are accurate. However, the recall is lower at 0.58, indicating the model correctly identifies 58% of all actual Hate Speech instances. The F1-score for Hate Speech is 0.68, reflecting a trade-off with higher precision than recall. The support value for this class is 90.The model's overall accuracy is 0.75, meaning it correctly predicts 75%

of the instances in the dataset. The macro average precision is 0.77, the macro average recall is 0.73, and the macro average F1-score is 0.74. These averages treat each class equally, regardless of their support values. The weighted averages, which consider the number of instances in each class, show a precision of 0.76, a recall of 0.75, and an F1-score of 0.74. The model performs well for Not Hate Speech, with a high recall of 0.89 indicating it correctly identifies most instances, although the precision of 0.72 suggests some false positives. For Hate Speech, the model has a higher precision of 0.81 but a lower recall of 0.58, indicating it correctly identifies fewer actual Hate Speech instances while being more confident in its predictions. The accuracy of 0.75 reflects good overall performance, correctly classifying 75% of instances. The macro and weighted averages provide balanced metrics that assess the model's overall performance across both classes, considering class imbalance. Overall, the model is effective at identifying Not Hate Speech but has room for improvement in detecting Hate Speech, as evidenced by the lower recall for Hate Speech.

4.2. Discussion

The first model and second model both demonstrate strong performance in detecting hate speech and non-hate speech, but with notable differences in their metrics. The first model shows a higher overall accuracy of 0.78 compared to the second model's 0.75. For the "Not Hate Speech" class, the first model achieves a recall of 0.84 and a precision of 0.77, while the second model has a higher recall of 0.89 but a lower precision of 0.72. This indicates that the second model is better at identifying non-hate speech instances but at the cost of more false positives. In the "Hate Speech" class, the first model achieves a precision of 0.78 and a recall of 0.70, resulting in an F1-score of 0.74. The second model, however, has a higher precision of 0.81 but a significantly lower recall of 0.58, leading to an F1-score of 0.68. This suggests that the second model is more confident in its hate speech predictions but misses more actual hate speech instances. Considering these results, if the primary goal is to identify as many hate speech instances as possible (minimizing false negatives), the first model is preferable due to its higher recall of 0.70. This model is better at catching most of the hate speech instances, even if it means slightly more false positives. Conversely, if the application requires high confidence in the predictions made (minimizing false positives), the second model is better due to its higher precision of 0.81. This model is more accurate in labeling instances as hate speech but misses more actual hate speech instances (lower recall).

Given the importance of both identifying hate speech accurately and ensuring high detection rates, the first model is generally the better choice for hate speech detection. Its higher recall and balanced F1-score suggest it is more effective at catching hate speech instances while maintaining reasonable precision. This balanced approach is often more desirable in applications where the cost of missing hate speech is high, such as in social media moderation or automated reporting systems.

5. Conclusion

The first model demonstrated a higher overall accuracy (0.78) compared to the second model (0.75). It achieved a balanced performance with a precision of 0.78, recall of 0.70, and F1-score of 0.74 for detecting hate speech. The second model, while having a higher precision of 0.81 for hate speech, had a significantly lower recall of 0.58, resulting in an F1-score of 0.68. The first model also performed well in identifying non-hate speech, with a precision of 0.77 and recall of 0.84.

This research contributes to the field of hate speech detection by providing a comparative analysis of two models combining Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). The findings highlight the trade-offs between precision and recall in hate speech detection, demonstrating the importance of balanced performance metrics in evaluating such models. This study offers insights into the effectiveness of different architectural choices and training strategies, including the use of L2 regularization and early stopping.

The results of this study have practical implications for the development of automated hate speech detection systems. The first model's balanced performance suggests it is more suitable for applications where the cost of missing hate speech is high, such as in social media moderation or automated reporting systems. The insights gained from this research can guide practitioners in selecting and fine-tuning models based on specific needs, whether prioritizing recall to minimize missed detections or precision to reduce false positives.

One limitation of this research is the use of a specific dataset for training and evaluation, which may not fully represent the diversity and complexity of real-world hate speech. Additionally, the models were trained for a fixed number of epochs with predefined hyperparameters, which might not be optimal for all datasets. The study also focused on text-based hate speech detection, not considering multimedia content such as images or videos, which are increasingly prevalent in online hate speech.

Future research should explore the generalization of these models to more diverse and comprehensive datasets to enhance their robustness and applicability. Additionally, experimenting with various hyperparameter optimization techniques and incorporating advanced regularization methods could enhance model performance. Finally, exploring explainable AI techniques to make hate speech detection models more transparent and understandable for users and moderators is a promising area for future work.

References

[1] J. Howarth, "17 Cyberbullying Facts & Statistics (2024)," Exploding Topics. Accessed: Jun. 06, 2024. [Online]. Available: https://explodingtopics.com/blog/cyberbullying-stats.

[2] G. Gambino and R. Pirrone, "CHILab @ HaSpeeDe 2: Enhancing hate speech detection with part-of-speech tagging," in *CEUR Workshop Proceedings*, 2020. doi: 10.4000/books.aaccademia.7057.

[3] M. Sajjad, F. Zulifqar, M. U. G. Khan, and M. Azeem, "Hate Speech Detection using Fusion Approach," in *2019 International Conference on Applied and Engineering Mathematics, ICAEM 2019 - Proceedings*, 2019. doi: 10.1109/ICAEM.2019.8853762.

[4] T. Parcollet *et al.*, "Quaternion recurrent neural networks," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[5] S. Riyadi, A. D. Andriyani, A. M. Masyhur, C. Damarjati, and M. I. Solihin, "Detection of Indonesian Hate Speech on Twitter Using Hybrid CNN-RNN," in *Proceeding - International Conference on Information Technology and Computing 2023, ICITCOM 2023*, 2023, pp. 352–356. doi: 10.1109/ICITCOM60176.2023.10442041.

[6] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural Computation*, vol. 31, no. 7. 2019. doi: 10.1162/neco_a_01199.

[7] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Networks*, vol. 106, 2018, doi: 10.1016/j.neunet.2018.07.011.
[8] M. Kamyab, G. Liu, and M. Adjeisah, “Attention-Based CNN and Bi-LSTM Model Based on TF-IDF and GloVe Word Embedding for Sentiment Analysis,” *Appl. Sci.*, vol. 11, no. 23, 2021, doi: 10.3390/app112311255.
[9] X. Xie, M. Xie, A. J. Moshayedi, and M. H. Noori Skandari, “A Hybrid Improved Neural Networks Algorithm Based on L2 and Dropout Regularization,” *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/8220453.
[10] C. Yu, P. H. Hung, J. H. Hong, and H. Y. Chiang, “Efficient Max Pooling Architecture with Zero-Padding for Convolutional Neural Networks,” in *GCCE 2023 - 2023 IEEE 12th Global Conference on Consumer Electronics*, 2023. doi: 10.1109/GCCE59613.2023.10315268.
[11] M. O. Ibrohim and I. Budi, “Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter,” *ALW3 3rd Work. Abus. Lang. Online*, pp. 46–57, 2019, [Online]. Available: https://www.aclweb.org/anthology/W19-3506.pdf
[12] Y. Setiawan, D. Gunawan, and R. Efendi, “Feature Extraction TF-IDF to Perform Cyberbullying Text Classification: A Literature Review and Future Research Direction,” in *2022 International Conference on Information Technology Systems and Innovation, ICITSI 2022 - Proceedings*, 2022, pp. 283–288. doi: 10.1109/ICITSI56531.2022.9970942.
[13] S. D. A. Putri, M. O. Ibrohim, and I. Budi, *Abusive Language and Hate Speech Detection for Indonesian-Local Language in Social Media Text*, vol. 251. 2021. doi: 10.1007/978-3-030-79757-7_9.
[14] R. Hendrawan, Adiwijaya, and S. Al Faraby, “Multilabel Classification of Hate Speech and Abusive Words on Indonesian Twitter Social Media,” in *2020 International Conference on Data Science and Its Applications, ICoDSA 2020*, 2020. doi: 10.1109/ICoDSA50139.2020.9212962.
[15] A. B. Syahputri and Y. Sibaroni, “Comparative Analysis of CNN and LSTM Performance for Hate Speech Detection on Twitter,” in *International Conference on ICT Convergence*, 2023, pp. 190–195. doi: 10.1109/ICoICT58202.2023.10262656.
[16] R. Larracy, A. Phinyomark, and E. Scheme, “Machine Learning Model Validation for Early Stage Studies with Small Sample Sizes,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2021. doi: 10.1109/EMBC46164.2021.9629697.
[17] A. Adli and P. Tyrrell, “Impact of Training Sample Size on the Effects of Regularization in a Convolutional Neural Network-based Dental X-ray Artifact Prediction Model,” *J. Undergrad. Life Sci.*, vol. 14, no. 1, 2020, doi: 10.33137/juls.v14i1.35883.
[18] F. Chollet and others, “Keras Tokenizer.” 2023.
[19] K. Mukai and T. Yamanaka, “Improving Translation Invariance in Convolutional Neural Networks with Peripheral Prediction Padding,” in *Proceedings - International Conference on Image Processing, ICIP*, 2023. doi: 10.1109/ICIP49359.2023.10223102.
[20] A. D. Nguyen, S. Choi, W. Kim, S. Ahn, J. Kim, and S. Lee, “Distribution Padding in Convolutional Neural Networks,” in *Proceedings - International Conference on Image Processing, ICIP*, 2019. doi: 10.1109/ICIP.2019.8803537.
[21] M. Sun, X. Yang, and Y. Xie, “Deep learning in aquaculture: A review,” *J. Comput*. csroc.org.tw, 2020. [Online]. Available: http://www.csroc.org.tw/journal/JOC31-1/JOC3101-28.pdf
[22] A. (2022). . Khan, “Balanced Split: A new train-test data splitting strategy for imbalanced datasets. ArXiv,” *arxiv.org*, 2022, [Online]. Available: https://arxiv.org/abs/2212.11116
[23] M. Zhu *et al.*, “Class weights random forest algorithm for processing class imbalanced medical data,” *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2789428.
[24] P. Wibowo and C. Fatichah, “An in-depth performance analysis of the oversampling techniques for high-class imbalanced dataset,” *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 7, no. 1, 2021, doi: 10.26594/register.v7i1.2206.
[25] V. M. González-Barcenas, E. Rendón, R. Alejo, E. E. Granda-Gutiérrez, and R. M. Valdovinos, “Addressing the Big Data Multi-class Imbalance Problem with Oversampling and Deep Learning Neural Networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019. doi: 10.1007/978-3-030-31332-6_19.
[26] O. Hrinchuk, V. Khrulkov, L. Mirvakhabova, E. Orlova, and I. Oseledets, “Tensorized embedding layers,” in *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, 2020. doi: 10.18653/v1/2020.findings-emnlp.436.
[27] F. Heimerl and M. Gleicher, “Interactive Analysis of Word Vector Embeddings,” *Comput. Graph. Forum*, vol. 37, no. 3, 2018, doi: 10.1111/cgf.13417.
[28] A. Templeton, “Word Equations: Inherently Interpretable Sparse Word Embeddings through Sparse Coding,” in *BlackboxNLP 2021 - Proceedings of the 4th BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2021. doi: 10.18653/v1/2021.blackboxnlp-1.12.
[29] S. Ariwibowo, A. S. Girsang, and Diana, “Hate Speech Text Classification Using Long Short-Term Memory (LSTM),” in *ICOSNIKOM 2022 - 2022 IEEE International Conference of Computer Science and Information Technology: Boundary Free: Preparing Indonesia for Metaverse Society*, 2022. doi: 10.1109/ICOSNIKOM56551.2022.10034908.
[30] K. Sanjar, A. Rehman, A. Paul, and K. Jeonghong, “Weight dropout for preventing neural networks from overfitting,” in *2020 8th International Conference on Orange Technology, ICOT 2020*, 2020. doi: 10.1109/ICOT51877.2020.9468799.
[31] J. Tong, Z. Wang, and X. Rui, “A Multimodel-Based Deep Learning Framework for Short Text Multiclass Classification with the Imbalanced and Extremely Small Data Set,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/7183207.
[32] S. Afaq and S. Rao, “Significance Of Epochs On Training A Neural Network,” *Int. J. Sci. Technol. Res.*, vol. 9, no. 06, 2020.