CUET_Binary_Hackers@DravidianLangTech EACL2024: Hate and Offensive Language Detection in Telugu Code-Mixed Text Using Sentence Similarity BERT

Salman Farsi, Asrarul Hoque Eusha
Jawad Hossain, Shawly Ahsan, Avishek Das and **Mohammed Moshiul Hoque**
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology, Chattogram-4349, Bangladesh
{salman.cuet.cse, asrar2860}@gmail.com
{u1704039, u1704057}@student.cuet.ac.bd, {avishek, moshiul_240}@cuet.ac.bd

Abstract

With the continuous evolution of technology and widespread internet access, various social media platforms have gained immense popularity, attracting a vast number of active users globally. However, this surge in online activity has also led to a concerning trend by driving many individuals to resort to posting hateful and offensive comments or posts, publicly targeting groups or individuals. In response to these challenges, we participated in this shared task. Our approach involved proposing a fine-tuning-based pre-trained transformer model to effectively discern whether a given text contains offensive content that propagates hatred. We conducted comprehensive experiments, exploring various machine learning (LR, SVM, and Ensemble), deep learning (CNN, BiLSTM, CNN+BiLSTM), and transformer-based models (Indic-SBERT, m-BERT, MuRIL, Distil-BERT, XLM-R), adhering to a meticulous fine-tuning methodology. Among the models evaluated, our fine-tuned L3Cube-Indic-Sentence-Similarity-BERT or Indic-SBERT model demonstrated superior performance, achieving a macro-average F1-score of 0.7013. This notable result positioned us at the 6^{th} place in the task. The implementation details of the task will be found in the GitHub repository [1].

1 Introduction

The contemporary digital landscape is heavily influenced by the pervasive role of social media in facilitating online communication. Platforms such as YouTube, Instagram, Facebook, and Twitter have not only provided users with avenues for creating and sharing content but have also become arenas where individuals can freely express their views and thoughts at any given moment (Taprial and Kanwar, 2012). The evolution of social media has brought forth a darker side, where individuals are defamed, targeted, and marginalized based on factors such as religion, physical appearance, or sexual orientation (Raja Chakravarthi et al., 2021). Given the impracticality of manually identifying offensive texts at scale, there arises a crucial need for an automated system capable of detecting hate speech. Such a system can empower relevant authorities to take necessary actions against offensive content. Natural Language Processing (NLP) emerges as a pivotal solution, offering various techniques to address these challenges effectively (Khurana et al., 2023). While the problem of identifying offensive language has been tackled from multiple angles, including detecting cyberbullying, aggression, toxicity, and abusive language (Fortuna et al., 2020; Mazari et al., 2023; Sharif et al., 2022; Hossain et al., 2022; Sharif and Hoque, 2021), there is a pressing need for more focused attention on hate-specific contexts in diverse languages.

Over the past few years, numerous studies have been conducted on detecting hate and offensive content in several high-resource languages such as English, Spanish, Arabic (Omar et al., 2020; Plaza-del Arco et al., 2021), and others that have ample linguistic resources, datasets, and related facilities. However, the challenge persists in addressing this issue efficiently for low-resource languages (Magueresse et al., 2020). In this particular task (B et al., 2024), the organizers presented a Telugu code-mixed hate speech dataset (Priyadharshini et al., 2023), framing it as a binary classification problem. The objective is to discern whether a given text represents any hate and offensive speech or not. This task serves as a crucial step toward addressing the gap in efficient hate speech detection for low-resource languages like Telugu, especially in the context of code-mixed text. As part of the participants in this task, the main contributions of our work are outlined below:

- We explored different ML, DL, and transformer-based models for hate speech de-

[1] https://github.com/Salman1804102/DravidianLangTech-EACL-2024-HOLD

Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 193–199
March 22, 2024 ©2024 Association for Computational Linguistics

tection. And boosted the model's performance by determining the optimal hyper-parameters.

- Contributed to the field by conducting a comprehensive comparison of different models and evaluating the performance of these models.

We organized the rest of our presentation as follows: section 2 delves into related work, section 3 describes the task and dataset, section 4 outlines our methodology, section 5 describes the experimental setup, section 6 presents the results analysis, section 7 conducts an in-depth error analysis, and finally, section 8 concludes with insights and outlines directions for future work.

2 Related Work

In the evolving landscape of hate and offensive text detection, researchers have explored a spectrum of techniques, each contributing to the continuous refinement of models. An influential Bengali abusive text detection endeavor was conducted (Eshan and Hasan, 2017) assessing the efficacy of Support Vector Machine (SVM), Random Forest (RF), and Naive Bayes (NB) classifiers. Their framework, achieving an accuracy of approximately 95%, laid a foundation for subsequent investigations into more advanced methodologies. Saumya et al. (2021) investigated offensive language identification in social media code-mixed Tanglish (Tamil+English) and Manglish (Malayalam+English) text, as well as Malayalam script-mixed. The N-gram TF-IDF-based MNB classifier achieved a weighted F1-score of 0.90 in Tamil code-mixed text whereas LR led with 0.78 for Malayalam code-mixed content. The Vanilla Neural Network (VNN) outperformed in handling Malayalam script-mixed text, achieving an impressive weighted F1-score of 0.95.

As the field matured, a notable shift emerged from traditional machine learning to deep learning, exemplified by Omar et al. (2020)'s work on the detection of Arabic hate speech. Using Recurrent Neural Networks (RNN), they achieved an exceptional 98.7% accuracy which outperformed Convolutional Neural Networks (CNN). Another study (Mazari et al., 2023) employed a multi-label approach for hate speech detection on social media, utilizing pre-trained BERT and ensemble learning architectures that include BiLSTM and BiGRU models. Integrating recent word embedding techniques and DL models, the proposed approach achieved a remarkable ROC-AUC score of 98.63%.

The exploration of transformer-based models added a layer of complexity to hate speech detection. A weighted ensemble technique (Sharif et al., 2022), incorporating m-BERT, Distil-BERT, and Bangla-BERT, demonstrated the adaptability of these models in handling diverse linguistic nuances, particularly in Bengali aggressive text datasets. In DravidianLangTech2021[2], the author (Sharif et al., 2021) addressed the challenge of detecting offensive text in code-mixed social media data, employing effective transformer-based models like XLM-R, m-BERT, and Indic-BERT for Tamil, Kannada, and Malayalam languages. Extending this exploration, another study (Saha et al., 2021) within the same task also delved into a diverse set of transformer-based models, including MuRIL, Distil-BERT, and others. In HASOC 2023[3], a study (Joshi and Joshi, 2023) evaluated the efficacy of various sentence-BERT models, including Bengali-SBERT, Gujarati-SBERT, Assamese-BERT, and L3Cube Indic-SBERT, showcasing state-of-the-art results in detecting hate speech within Indian linguistic contexts.

3 Task and Dataset Description

In this shared task (B et al., 2024), a Telugu code-mixed dataset was introduced for the detection of hate and offensive language (Priyadharshini et al., 2023). The dataset, designed for binary classification, comprises diverse social media posts and comments containing both hate/offensive text and non-hate/non-offensive text. For participants, both the training and test datasets were provided, without any separate validation set. The training dataset consisted of 4,000 samples, comprising 2,061 non-hate-labeled and 1,939 hate-labeled samples, demonstrating a well-balanced distribution. Some other useful insights are mentioned in Table 1.

Set	Class	Sample Count	UW	MxL	AL	OOV
Train	Hate	2,061	17,097	71	10	1,167
	Non-Hate	1,939				
Test	Hate	1,939	2,365	18	7	
	Non-Hate	1,939				

Table 1: Dataset statistics, including UW (unique words), MxL (maximum length), AL (average length), and OOV (out-of-vocabulary) words in texts

[2] https://dravidianlangtech.github.io/2021/index.html

[3] https://hasocfire.github.io/hasoc/2023/

4 Methodology

In this section, we will delineate our methodology step by step. Figure 1 shows a schematic diagram of the methodology.

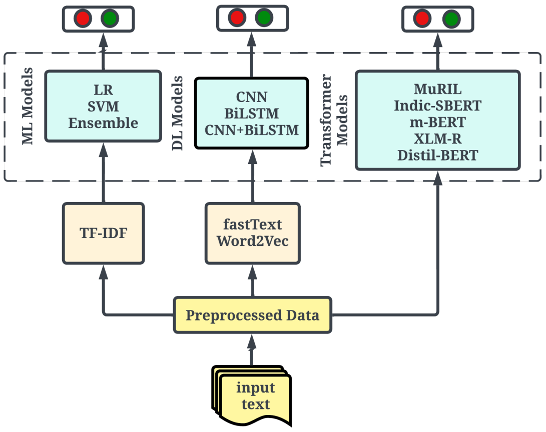

Figure 1: A schematic diagram of the methodology.

4.1 Data Pre-processing

Given that the dataset is code-mixed and sourced from social media, it inherently includes a substantial amount of extraneous and redundant content. Therefore, as an initial step, we conducted thorough data pre-processing. This involved the removal of emojis, symbols, signs, numbers, and certain unnecessary punctuation marks from the text.

4.2 Feature Extraction

In selecting feature extraction methods, our rationale is rooted in enhancing the interpretability and efficiency of ML and DL models for text data comprehension. TF-IDF was chosen for ML to capture important unigram features and highlight their significance in the context of our study (Das et al., 2023). For DL models, fastText and Word2Vec were employed to harness semantic relationships and context within the text. The implementation choices, such as the dimensionality of 300 for both Word2Vec and fastText, were made to strike a balance between computational efficiency and representation effectiveness, as supported by existing literature (Bojanowski et al., 2017; Mikolov et al., 2013). This approach ensures a comprehensive understanding of the textual content by both ML and DL models.

4.3 ML Models

To identify instances of hate speech, our initial approach involved the utilization of fundamental machine learning (ML) models. Specifically, we employed LR, SVM, and subsequently applied an ensemble technique incorporating RF, LR, SVM, and Decision Tree (DT) (Sarker, 2021). To train the LR model, we selected ‘liblinear’ as the solver, and set the parameter value of C to 1. For SVM, ‘sigmoid’ was chosen as the optimizer, and the C value was set to 1. This systematic deployment of basic ML models and an ensemble approach formed the initial exploration of our hate speech detection task.

4.4 DL Models

To leverage the proven efficacy of deep learning (DL) methods in handling sequence data, we incorporated three distinct approaches: Bidirectional Long Short Term Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997), Convolutional Neural Network (CNN) (O’Shea and Nash, 2015) and a combination of CNN and BiLSTM (Sharif and Hoque, 2021). Each of these models was trained with both fastText and Word2Vec embeddings. The CNN model includes a 1D convolutional layer with 128 filters and a kernel size of 5, followed by global max pooling for feature extraction.

Meanwhile, in our combined CNN + BiLSTM (Khan et al., 2022) methodology, the CNN layer processed the initial embedding features using 128 filters. Subsequently, a max-pooling operation with a window size of 2 was applied to distill relevant features. The resultant vector underwent processing in the BiLSTM layer, which featured 200 bidirectional cells to adeptly capture long-term dependencies. To address overfitting concerns, a dropout technique with a 0.2 rate was implemented in the BiLSTM layer. The final step involved feeding the concatenated output of the BiLSTM layer into a sigmoid layer for prediction.

4.5 Transformer-based Models

Transformer-based models, particularly the latest addition preceding GPT, have revolutionized text classification and various problem domains (Gasparetto et al., 2022). Our method capitalizes on the versatility of pre-trained transformer-based models, evaluating their performance across different hyper-parameters. All the transformer-based models were trained using ktrain (Maiya, 2022) and imported from the ‘Hugging Face’[4] (Wolf et al., 2019) library by incorporating a random seed for

[4] https://huggingface.co/

result reproducibility. Specifically, we employed m-BERT (Devlin et al., 2019), Distil-BERT (Sanh et al., 2019), MuRIL (Sakorikar et al., 2021), Indic-SBERT (Deode et al., 2023), and XLM-R (Conneau et al., 2020).

L3Cube Indic-SBERT, a multilingual Sentence-BERT model, is customized for Indian languages through fine-tuning vanilla BERT (Gao et al., 2019) models with a synthetic corpus. Demonstrating outstanding cross-lingual performance, it outperforms alternatives like LaBSE (Feng et al., 2020) and LASER (Artetxe and Schwenk, 2019) in sentence similarity tasks across diverse Indian languages, providing a valuable resource for natural language understanding in the Indian multilingual context.

5 Experimental Setup

The hyper-parameters used in this task were determined through an iterative process involving frequent trials. The choice of the parameters depicted in Table 2 also aligns with common practices in binary classification tasks for DL models (Plested et al., 2021; Roy et al., 2023). On the other hand, the hyper-parameters for transformer-based models are shown in Table 3. This meticulous experimentation aimed to optimize model performance, ensuring the chosen hyperparameters strike a balance between convergence and computational efficiency. The consistent application of these settings across all models facilitates a fair and meaningful comparison, allowing us to isolate the impact of architectural variances on overall performance.

Parameter	Value
Optimizer	Adam
Loss Function	Binary Crossentropy
Activation (Hidden Layer)	ReLU
Activation (Output Layer)	Sigmoid
Learning Rate	$1e^{-3}$
Batch Size	32
Epochs	30
MaxLen	80
Dropout	0.2

Table 2: Experimental setup for the DL models.

Parameter	Value
Optimizer	AdamW
Learning Rate	$3e^{-5}$
Batch Size	16
Maxlen	100
Epochs	10

Table 3: Experimental setup for the transformer-based models.

6 Result Analysis

The results in Table 4 unveil significant patterns and challenges across the evaluated models. In the ML category, LR and SVM classifiers demonstrate competitive precision, recall, and F1 scores, with SVM achieving the highest F1-score of 0.65. However, the ensemble method, while achieving a comparable F1 score, shows slightly lower precision and recall, suggesting potential challenges in integrating diverse ML models. Moving to DL models, those utilizing fastText embeddings consistently outperform Word2Vec counterparts. The best-performing model using Word2Vec embeddings was the hybrid CNN+BiLSTM, achieving an F1-score of 0.42. In contrast, the BiLSTM model achieved an F1-score of 0.60 using fastText word embeddings. The stark difference in F1-score among models using these two embeddings may be attributed to Word2Vec's struggle to capture the rich semantic information present in code-mixed text. The intricacies of code-mixing, where multiple languages coexist, pose a challenge for traditional embeddings, impacting their ability to represent nuanced meanings effectively.

Methods	Classifiers	P	R	F1
ML	LR	0.63	0.63	0.63
	SVM	0.65	0.65	0.65
	Ensemble	0.60	0.60	0.59
DL	CNN(Word2Vec)	0.54	0.51	0.40
	BiLSTM(Word2Vec)	0.58	0.52	0.41
	CNN+BiLSTM(Word2Vec)	0.56	0.52	0.42
	CNN(fastText)	0.64	0.60	0.57
	BiLSTM(fastText)	0.68	0.63	0.60
	CNN+BiLSTM(fastText)	0.65	0.60	0.55
TransF	m-BERT (uncased)	0.65	0.65	0.65
	m-BERT (cased)	0.69	0.69	0.69
	MuRIL	0.68	0.69	0.69
	XLM-R	**0.70**	0.69	**0.70**
	Distil-BERT	0.67	0.67	0.67
	Indic-SBERT	**0.70**	**0.70**	**0.70**

Table 4: Result comparison on test data where P, R and F1 denote precision, recall, macro F1-score and TransF denotes transformer-based model.

However, transformer-based models exhibited promising performance compared to both ML and DL models. XLM-R and Indic-SBERT both achieved the highest F1 score of 0.70. Due to the higher recall value of 0.70 in the case of Indic-SBERT, it was selected as the best model for our task, showcasing its adaptability to the complexities of code-mixed language. Table 5 illustrates the impressive performance of this model, among the other participating teams.

Team Name	Run	F1-Score	Rank
Sandalphon	1	0.7711	1
Selam	2	0.7711	2
CUET_Binary_Hackers	**2**	**0.7013**	**6**
MUCS	3	0.6501	15

Table 5: A brief ranking of participating teams.

In summary, DL models performed less effectively compared to ML and transformer-based models. The reason for this weaker performance is the extensive appearance of cross-lingual words in the text. As a result, Word2Vec and fastText embeddings failed to create appropriate feature mappings among the words (Sharif et al., 2021). Thus, LSTM and CNN-based models may not have found sufficient relational dependencies among the features, performing below expectations. However, Indic-SBERT outperformed other models, due to its ability to capture intricate semantic relationships and contextual nuances inherent in the language mixture. Sentence-BERT models, like Indic-SBERT, excel in understanding the semantic similarity between sentences, making them well-suited for code-mixed text comprehension. The model's robust encoding of semantic information enables it to effectively navigate the intricacies of Telugu code-mixing, contributing to its superior performance in this specific linguistic context.

7 Error Analysis

To comprehensively analyze the performance of L3Cube Indic-SBERT, we provide a detailed error analysis in this section, utilizing a confusion matrix depicted in Figure 2. Out of 250 samples, 179 hate speech and 172 non-hate speech samples were correctly classified. However, there were 71 misclassified hate speech samples and 78 misclassified non-hate speech samples. The misclassification rates for both hate and non-hate classes are 28.4% and 31.2% respectively. The minimal difference suggests a close misclassification rate between the two labels, potentially influenced by slight variations in the number of types of training samples.

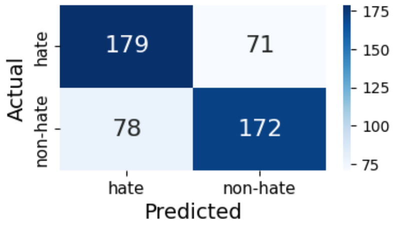

Figure 2: Confusion matrix for the Indic-SBERT model.

Additionally, similar code-mixed words between the two classes may contribute to this issue when the model attempts to understand the text's meaning. To reduce misclassification, a detailed analysis of each word in misclassified samples using the Named Entity Recognition (NER) method can be done to remove redundant codemixed words.

Limitations

- Our work relies on pre-trained transformer-based models, which may pose challenges in scenarios where the context significantly deviates from the model's training data.
- The employed DL models didn't perform well. It requires further investigation using other embeddings and building better models.
- GPU limitations hindered us from experimenting with the ensemble of transformers.

8 Conclusion and Future Work

This paper delves into the exploration and evaluation of various ML, DL, and transformer-based approaches. Our initial investigation involved TF-IDF and embedding features (Word2Vec & fastText), followed by systematic experiments with ML and DL methods. The results indicate that SVM outperformed other ML and DL models with an F1-score of 0.65. However, incorporating the transformer model significantly enhanced overall performance. Specifically, Indic-SBERT, stood out by achieving the highest F1-score of 0.70. Future exploration can involve incorporating contextualized embeddings like GPT, ELMO, and FLAIR, or experimenting with ensembling transformers and fusion models tailored to hate speech contexts. Besides, alternative embedding techniques such as GloVe and BERT-based embeddings can be applied to enhance the performance of DL models.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Premjith B, Bharathi Raja, Prasanna Kumar Kumaresan, Saranya Rajiakodi, Sai Prashanth Karnati, Sai Rishith Reddy Mangamuru, and Janakiram Chandu. 2024. Findings of the Shared Task on Hate and Offensive Language Detection in Telugu Codemixed Text (HOLD-Telugu). In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, Malta. European Chapter of the Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Alexis Conneau, Kartik Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Udoy Das, Karnis Fatema, Md Ayon Mia, Mahshar Yahan, Md Sajidul Mowla, Md Fayez Ullah, Arpita Sarker, and Hasan Murad. 2023. EmptyMind at BLP-2023 Task 1: A Transformer-based Hierarchical-BERT Model for Bangla Violence-Inciting Text Detection. In *Proceedings of the First Workshop on Bangla Language Processing (BLP-2023)*, pages 174–178, Singapore. Association for Computational Linguistics.

Samruddhi Deode, Janhavi Gadre, Aditi Kajale, Ananya Joshi, and Raviraj Joshi. 2023. L3Cube-IndicSBERT: A simple approach for learning cross-lingual sentence representations using multilingual BERT. *arXiv preprint arXiv:2304.11434*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.

Shahnoor C Eshan and Mohammad S Hasan. 2017. An application of machine learning to detect abusive Bengali text. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pages 1–6. IEEE.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *arXiv preprint arXiv:2007.01852*.

Paula Fortuna, Juan Soler, and Leo Wanner. 2020. Toxic, hateful, offensive or abusive? what are we really classifying? an empirical analysis of hate speech datasets. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6786–6794, Marseille, France. European Language Resources Association.

Zhengjie Gao, Ao Feng, Xinyu Song, and Xi Wu. 2019. Target-dependent sentiment classification with BERT. *Ieee Access*, 7:154290–154299.

Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. 2022. A survey on text classification algorithms: From text to predictions. *Information*, 13(2):83.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Alamgir Hossain, Mahathir Bishal, Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. COMBATANT@TamilNLP-ACL2022: Fine-grained categorization of abusive comments using logistic regression. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 221–228, Dublin, Ireland. Association for Computational Linguistics.

Ananya Joshi and Raviraj Joshi. 2023. Harnessing Pre-Trained Sentence Transformers for Offensive Language Detection in Indian Languages. *arXiv preprint arXiv:2310.02249*.

Shakir Khan, Mohd Fazil, Vineet Kumar Sejwal, Mohammed Ali Alshara, Reemiah Muneer Alotaibi, Ashraf Kamal, and Abdul Rauf Baig. 2022. BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection. *Journal of King Saud University-Computer and Information Sciences*, 34(7):4335–4344.

Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2023. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, 82(3):3713–3744.

Alexandre Magueresse, Vincent Carles, and Evan Heetderks. 2020. Low-resource languages: A review of past work and future challenges. *arXiv preprint arXiv:2006.07264*.

Arun S Maiya. 2022. ktrain: A low-code library for augmented machine learning. *The Journal of Machine Learning Research*, 23(1):7070–7075.

Ahmed Cherif Mazari, Nesrine Boudoukhani, and Abdelhamid Djeffal. 2023. BERT-based ensemble learning for multi-aspect hate speech detection. *Cluster Computing*, pages 1–15.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Ahmed Omar, Tarek M Mahmoud, and Tarek Abd-El-Hafeez. 2020. Comparative performance of machine learning and deep learning algorithms for Arabic hate speech detection in osns. In *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)*, pages 247–257. Springer.

Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Flor Miriam Plaza-del Arco, M Dolores Molina-González, L Alfonso Urena-López, and M Teresa Martín-Valdivia. 2021. Comparing pre-trained language models for Spanish hate speech detection. *Expert Systems with Applications*, 166:114120.

Jo Plested, Xuyang Shen, and Tom Gedeon. 2021. Rethinking binary hyperparameters for deep transfer learning. In *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part II 28*, pages 463–475. Springer.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Malliga S, Subalalitha Cn, Kogilavani S V, Premjith B, Abirami Murugappan, and Prasanna Kumar Kumaresan. 2023. Overview of Shared-task on Abusive Comment Detection in Tamil and Telugu. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 80–87, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Bharathi Raja Chakravarthi, Dhivya Chinnappa, Ruba Priyadharshini, Anand Kumar Madasamy, Sangeetha Sivanesan, Subalalitha Chinnaudayar Navaneethakrishnan, Sajeetha Thavareesan, Dhanalakshmi Vadivel, Rahul Ponnusamy, and Prasanna Kumar Kumaresan. 2021. Developing Successful Shared Tasks on Offensive Language Identification for Dravidian Languages. *arXiv e-prints*, pages arXiv–2111.

Sunita Roy, Ranjan Mehera, Rajat Kumar Pal, and Samir Kumar Bandyopadhyay. 2023. Hyperparameter Optimization for Deep NeuralNetwork Models: A Comprehensive Study onMethods and Techniques.

Debjoy Saha, Naman Paharia, Debajit Chakraborty, Punyajoy Saha, and Animesh Mukherjee. 2021. Hate-alert@DravidianLangTech-EACL2021: Ensembling strategies for transformer-based offensive language detection. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 270–276, Kyiv. Association for Computational Linguistics.

Tushar Sakorikar, Pushpak Bhattacharyya, Surya Jauhar, and Mohit Neogi. 2021. MuRIL: Multilingual Representations for Indian Languages. *arXiv preprint arXiv:2103.09974*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Association for Computational Linguistics.

Iqbal H Sarker. 2021. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):160.

Sunil Saumya, Abhinav Kumar, and Jyoti Prakash Singh. 2021. Offensive language identification in Dravidian code mixed social media text. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 36–45, Kyiv. Association for Computational Linguistics.

Omar Sharif and Mohammed Moshiul Hoque. 2021. Identification and classification of textual aggression in social media: Resource creation and evaluation. In *International Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation*, pages 9–20. Springer.

Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2021. NLP-CUET@DravidianLangTech-EACL2021: Offensive language detection from multilingual code-mixed text using transformers. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 255–261, Kyiv. Association for Computational Linguistics.

Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2022. M-BAD: A multilabel dataset for detecting aggressive texts and their targets. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 75–85, Dublin, Ireland. Association for Computational Linguistics.

Varinder Taprial and Priya Kanwar. 2012. *Understanding social media*. Bookboon.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.