Improving Detection of Hate Speech, Offensive Language and Profanity in Short Texts with SVM Classifier

Surya Agustian[1], Zaky Idhafi[2] and Agit Fadillah Rihardi[3]

[1,2,3] *UIN Sultan Syarif Kasim, Jl. H.R. Soeberantas km 11.5 Panam, Pekanbaru, Riau, Indonesia*

Abstract

Hate speech and offensive language in social media have become a global issue, affecting various nations and languages. Conflicts on social media, triggered by hate speech and offensive language, can lead to victims experiencing mental health problem, disruptions in their peace, and disturbances in their real-world social lives. HASOC 2023 organizes some shared tasks to detect hate speech and offensive language in several languages spoken on the Indian peninsula, which categorized as low-resource languages. Tasks 1A and 1B in Sinhala and Gujarati languages conceal underlying difficulties, requiring the use of particular techniques in the classification procedure. This study proposes an SVM classifier method with improvement strategies for optimization and feature selection based on FastText word embeddings. The experimental results indicate that the applied strategies significantly enhance performance compared to the baseline method. The improvement achieved for Sinhala is 5.37%, and for Gujarati, it is 26.08% over the baseline method which use bag-of-words input features.

Keywords

SVM classifier, FastText word embeddings, optimization, feature selection

1. Introduction

Social media offers individuals the freedom to express their thoughts and emotions on a wide range of daily life issues. Unfortunately, this freedom is frequently misused to propagate hate speech, offensive language and profanity, even targeting people, group, and governments. Hate speech and offensive language remains a global concern on social media, regardless of the language spoken, as long as the internet and mobile phones are accessible. Hate speech and offensive language can arise from the differences in personal view or group opinions regarding religion, politics, ideology, social issues, gender, ethnicity, culture, economics, and more. Smedt et al. [1] investigated them in several domains and languages and found that across various topics, they exhibited similar characteristics of hateful expression.

Social media platforms such as Facebook, Instagram, Twitter, YouTube comment sections, and various community forums have become virtual battlegrounds where hatred and profanity are frequently unleashed. Bullying, whether by an individual or a group, is also often targeting other people or specific groups, which can lead to victims experiencing depression, stressed, and in some cases, may even result in suicide [2]. Therefore, messages contain hate speech and profane words need to be minimized, filtered, and removed from social media.

Various research and shared tasks to detect hate on social media have been carried out in various languages, such as Portuguese [3], Spanish [4], Arabic [5], Vietnamese [6], Italian [7] and Bahasa Indonesia [8]. Hate speech detection is also discussed on multilingual texts [9], [10], as well as on its level and target objects [9], [10].

Forum for Information Retrieval Evaluation, December 15-18, 2023, Goa, India

EMAIL: surya.agustian@uin-suska.ac.id (S. Agustian); 11950110021@students.uin-suska.ac.id (Z. Idhafi); 11751100389@students.uin-suska.ac.id (A.F. Rihardi)
ORCID: 0000-0002-9631-8237 (S. Agustian)

In addition to widely spoken languages, the hate speech classification task has piqued researchers' interest in languages from regions with limited resources. Since 2019, HASOC has organized shared tasks aimed at detecting hate speech and abusive language in various languages, including English, German, and the Indo-Aryan language family of the Indian subcontinent [11], [12]. At HASOC 2023, the shared tasks involve detecting hate speech in low-resource languages like Sinhala, Gujarati, Bengali, Bodo, and Assamese. Additionally, there is a task focused on identifying conversational hate speech in mixed languages, a continuation from the previous year, as well as a task for detecting hate speech spans within sentences [13].

Lexicon-based and machine-learning approaches were used in previous work [14] to detect hate speech in Sinhala. The lexicon list was generated through the translation of prohibited words in English into Sinhala. Apart from that, the source of profane and offensive words was also obtained from online sources, and then various variations were taken through the dataset collection for this research. The Convolution Neural Networks (CNNs) method in [15] is used to detect hate speech in Sinhala language. The first CNN model is trained to detect the presence or absence of hate speech, then if it is detected, the level of hate speech will be detected by a second CNN model, which is trained separately to classify the level of hate speech. Gujarati is also known as a low resource language. Hate speech detection in Gujarati in [16] still uses the external resource of a list of sentiment words from word-net.

There are still numerous challenges in hate speech detection research, which is why this task continues to capture researchers' interest in search of the most optimal automatic solution. Among these challenges are the subjective nature of determining which sentences contain hate speech—where the context of the conversation often determines whether a sentence qualifies—and the limited availability of data, among others [17], [18]. In HASOC 2023 Task 1, besides facing the constraint of being a low-resource language, we identified a fundamental issue with the dataset for the Sinhala language—namely, an imbalance in the number of samples between the HOF (Hate, Offensive, and Fear) and NOT classes within the coarse-grained text data. On the other hand, for Gujarati, the scarcity of training data poses a significant challenge, making it difficult to construct optimal models for various detection methods using machine learning.

Due to limited knowledge and references for these two languages, we rely solely on the robustness of the proposed Machine Learning method. We adopt the SVM method with word embeddings as input features, following a similar approach used for hate speech detection in Indonesian tweets [19], with specific optimizations addressing these challenges. We opted not to use a transformer-based method due to our limited proficiency in these languages, which posed challenges when creating an appropriate training dataset compatible with the pre-trained BERT model.

The rest of this paper is organized in this sequence: Section 2 will disucss about the proposed method in Task 1A and 1B. The performance improvement of the baseline method as the results of optimization is described in section 3. The final section will summarize this finding and offer suggestions for further works.

2. Research Methodology

The Task 1 of HASOC 2023 is focused on hate speech and offensive language detection in Sinhala (Task 1A) and Gujarati (Task 1B). Sinhala, one of the Indo-Aryan languages, is considered as a language with limited resources. It is spoken by more than 17 million individuals in Sri Lanka and holds the status of Sri Lanka's official languages. Gujarati as well, classified as a low-resource Indo-Aryan language, has approximately 50 million native speakers and remain one of the 22 official languages recognized in India. Both tasks are binary classification with label HOF (Hate and Offensive) if the post containing hate, offensive and profane language, and NOT for Non Hate-Offensive class.

The statistic of the dataset provided for training and development is describe in Table 1 below. In Sinhala [20], the total number of posts sufficient to train a machine learning, i.e. 7500 posts, but for Gujarati, the size of dataset for training is very small,which is only 200 post. Regarding the compositions, Gujarati has a same number of sample of each class, while Sinhala is imbalanced, as describe in Table 1. In our development phase, we split the datasets into train and validation sets, with proportion of 90:10.

Table 1
Label distribution of Sinhala and Gujarati dataset for training

Language	Label	Number of posts	Percentile
Sinhala (7500 posts)	HOF	3176	42.35%
	NOT	4324	57.65%
Gujarati (200 posts)	HOF	100	50%
	NOT	100	50%

2.1. SVM Method

Support Vector Machine (SVM) is a machine learning method that can be used for classification tasks, by finding a hyperplane that separates two classes of data. This hyperplane is the optimal decision boundary that maximizes the margin (distance) between the two classes. SVM aims to find this hyperplane with maximum margin. With this technique, it performs remarkably well when categorizing data into two classes efficiently. SVM stands as a cutting-edge machine learning algorithm that was initially designed for solving binary classification challenges, and later enhanced to tackle multiclass classification problems and regression tasks.

We proposed SVM as a state-of-the-art text classifier [21] for Task 1A and 1B. Both are first classified with bag of words feature set as baseline method. For our baseline, we employ the tokenizer from scikit-learn[2] to extract both word unigrams and bigrams from each tweet, which are then transformed into TF-IDF features vector. In the case of Sinhala, it results in a vocabulary list comprising the substantial 47,316 n-grams, and 2,083 n-grams in Gujarati. Due to a lack of knowledge in the Sinhala and Gujarati language, we utilize all n-grams as input features, resulting in a baseline SVM input dimensionality of 47,316 and 2,083 respectively. However, this approach produced a very wide level of sparsity in each sentence vector, specifically for Sinhala, as it utilize the entire vocabulary to form a vector. This condition is well-tackled by SVM with its robustness sparse technique and good generalization ability, which make SVM become a popular approach for supervised learning [22].

We employ basic text preprocessing before tokenize the tweets, i.e. removing numbers, punctuations, mentions, URLs, hashtags, retweet states like “RT @username:” and adding space between emojis. Stemming is not implemented based on the hypothesis that it might inadvertently reduce or strip away emotional nuances from written expressions in social media. Differently, stopwords can be treated as options. In some cases, retaining stopwords in tweets can be advantageous, but in others, their presence may lower accuracy if they are not removed.

2.2. Feature Selection

In general, many studies have reported the use of bag-of-words vector features as inputs for SVM text classifiers. While these methods have shown good performance among reported machine learning approaches, they can be inefficient in terms of computing and memory usage. We hypothesize that employing word embeddings as a feature could significantly reduce input vector dimensionality for various Machine Learning methods, including SVM. This word embeddings may not only enhance classification capabilities in terms of computing time but also improve classification performance (accuracy, precision, recall and F1-score). This is because converting sentences into vectors with smaller dimensions using word embeddings makes similarity measure between sentences more efficient. In contrast, bag-of-words vectors exhibit high sparsity.

[2] https://scikit-learn.org/

For example, let's consider two sentences with similar meanings but different words (synonyms), which may result in a low similarity score:

Sentence 1: “This research uses formula 2.5 to determine the direction of objects movements”

Sentence 2: “Our study employ equation 2.5 to calculate the direction of a moving object.”

Assuming stopwords are removed from the text, only the word 'direction' is common to both sentences. If stemming is applied, the word 'object' can be added to the set of shared words between sentence 1 and sentence 2. When using similarity measures like Jaccard or cosine similarity with bag-of-words vectors, the similarity results tend to be quite low because they rely solely on matched words. In contrast, when using word embeddings for similarity measurement, we believe the results will be better.

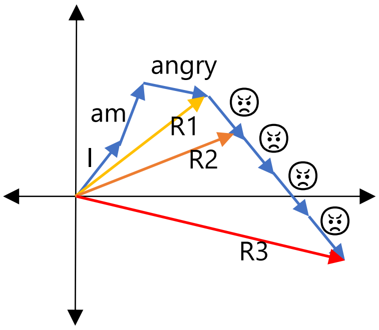
Based on this hypothesis, we use word embeddings as feature selection. Among word2vec [23], glove [24] and fasttext [25], we chose fasttext word embeddings with the hypothesis that fasttext can predict unseen words in generating word embeddings model (OOV, out of vocabulary). FastText can generate word vectors based on the composition of it's character n-grams, so that it can deal with unseen words during the training.

2.3. Optimization

The optimization steps undertaken in tasks 1A and 1B were tailored to address the underlying issues which we identified. In the case of Sinhala, the primary issue was the imbalance between the HOF and NOT classes. To tackle this class imbalance, various approaches were available, including oversampling the class with fewer instances or undersampling the class with more instances. We tend to use oversampling approach, as undersampling could lead to the loss of numerous word-specific features associated with the dominant class.

In the Sinhala dataset, as is typical in real-world scenarios, the neutral class tends to be more prevalent in social media posts which collected through keyword-based crawling. Given the smaller volume of data in the HOF class, we performed oversampling by randomly selecting existing tweets from the HOF class, where each tweet should contain more than 18 clean tokens. Emojis, when detected, were treated as individual tokens. Notably, during our analysis of the existing Sinhala tweet data, we did not find the use of emojis. However, in Gujarati, emojis are commonly employed, making it imperative to consider them due to the limited number of samples.

We employ a special treatment for emoji tokenization, including instances of multiple identical emojis in a row (e.g. emojis in text: “I am angry 😡 😡 😡 😡 ”). Instead of remove duplication, we treat each duplicate emoji as a separate token when generating sentence vector. This approach is based on the idea that emojis occupy distinct vectors within the word embedding vector space, and the repetition of each emoji influences the sentence's position in that space. Certain emojis can accentuate the emotional content of a tweet. For example, a sequence of angry emoji may indicate an emotional outburst by the author. By extracting duplicated emojis into single tokens, each will contribute in steering the sentence vector toward a particular direction, as illustrated in Figure 1. This direction will be more aligned and closer to the common sentiment or nuance associated with the feelings expressed by those emojis, either positive or negative. Furthermore, we also excluded stop words from the data used for oversampling in the Sinhala language.



Text: I am angry 😡 😡 😡 😡
Arrows are vectors in space
R : sentence vector
R1: emojis are removed
R2: duplicates are removed
R3: duplicates are taken into account

Figure 1. An illustration of how duplicate emojis steer sentence vector into a certain direction.

Sentence embeddings are computed from the resulting vector of word embeddings in normalized form, derived from the cleaned constituent tokens of a tweet, following equations (1) and (2). If V_t represents the word embedding vector for a token, given as $V_t = [v_1, v_2, \ldots v_d]$ with dimension=d, then the normalized vector V_n is obtained by element-wise division of the elements of V_t by the norm vector, as defined in equation (1). On the other hand, the sentence vector (V_s), composed of j tokens, is calculated by taking the element-wise average of the normalized vectors (V_n) of the tokens comprising the sentence, as illustrated in equation (2).

$$V_n = \frac{V_t}{\sqrt{\sum_i v_i{}^2}} \tag{1}$$

$$V_s = \frac{1}{j}\sum\nolimits_j V_{n\,j} \tag{2}$$

Another optimization step involved normalizing the sentence embeddings vector, as it contained elements with both negative or some values exceeding 1. To achieve this, we applied normalization using the scaling function from scikit-learn, which brings the magnitudes of vector elements converge to a range between 0 and 1. In this optimization process, we explored the best model performance using various normalization techniques, including min-max scaling, robust scaling, and no scaling, as discussed by Zikri and Agustian [19].

Data balancing for the Sinhala language is exclusively applied to the portion of the train dataset, which has been initially split into a 90:10 ratio for training (data-train) and validation (data-dev). This process yields data-train comprising 7,796 tweets, with each class containing 3,898 tweets. In contrast, the validation data (data-dev) remains unchanged. This approach simplifies the selection of the optimal SVM model after undergoing several improvement stages (baseline, feature selection, optimization). The training outcomes of each method (SVM models) are assessed to predict the data-dev. The model with highest F1-score (optimal model) is chosen to predict the testing data for our RUN submission.

In addition to the optimizations mentioned above, we also employ a grid search to obtain the optimal SVM parameters. This includes selecting the appropriate kernel (RBF, linear, or sigmoid) as well as determining optimal values for C and gamma, using 5-fold cross-validation.

3. Experiment and Results

We have designed a two-phase hate speech detection system, similar with the approach adopted by Agustian et al. [26], where we utilize an SVM classifier as the core machine learning component. The system's workflow is illustrated in Figure 2 below. Phase 1 focuses on building a language model using FastText word embeddings, whereas Phase 2 employs the SVM classifier to detect hate speech. The Python code for this method is made available on GitHub[3].

3.1. Experiment Setup

The tweet source in whole provided dataset flows into a set of preprocessing step before trained by FastText. It is optional to use or discard one or more steps in the below preprocessing set, to get improvement of classification results. We empirically choose the suitable steps and compare the prediction results on validation data (data-dev).

- Remove numbers
- Remove punctuation
- Remove words repetition
- Remove stopwords
- Replace mentions (@user, @USER, @AUTHOR) into single token “MENTIONED”
- Remove double spaces

[3] https://github.com/s4gustian/HASOC2023.git

- Remove URLs
- Tokenize (get every emojis into single tokens)
- Remove Latin words

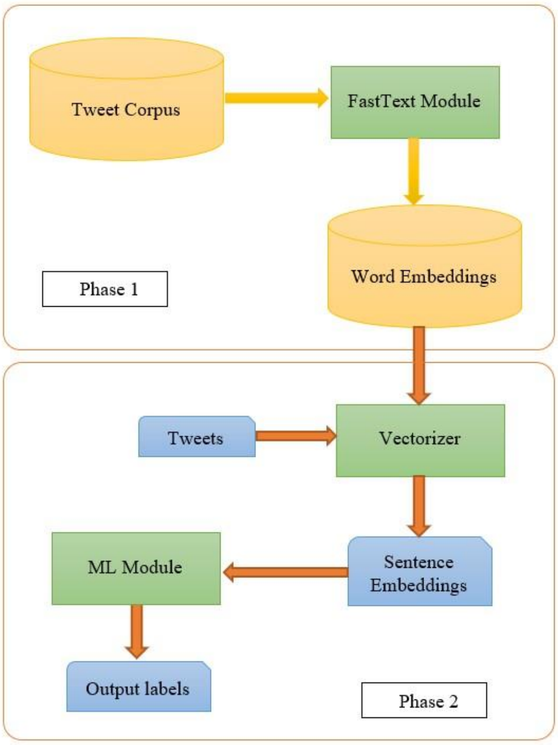


Figure 2. Two phase hate speech detection method [26]

The same preprocessing steps are applied to both data-train and data-dev sets before converting them into sentence embeddings. These vectors are then used as input features for SVM and undergo an optimization process, which may include normalization or without normalization. To address the imbalance of the classes in the data, oversampling techniques, as explained in the previous section, are applied. We conducted experiments using all combinations of these preprocessing and optimization techniques to identify the optimal model. The selected model is the one that achieved the highest F1-score on the validation data (data-dev) during training. Table 2 displays the experimental combinations conducted in our search for the optimal model, which we submitted to the HASOC 2023 system.

Table 2
Experiment Setup

Task	RUN	Feature	Balancing	Scaling	Single Emoji as token
Task 1A (Sinhala)	RUN1	Bag of Word	No	No	No
	RUN2	FastText	No	Yes	No
	RUN3	FastText	Yes	Yes	No
Task 1B (Gujarati)	RUN1	Bag of Word	No	No	No
	RUN2	FastText	No	Yes	No
	RUN3	FastText	No	Yes	Yes

3.2. Result and Discussion

The results from the conducted experiments reveal a significant improvement in each run, as shown in Table 3 below. The selection of FastText as an input feature not only speeds up computation due to its substantial reduction in vector dimensionality but also yields superior performance (RUN2).

Specifically, for Sinhala data, there was a notable increase of 4.96% in terms of the F1-score, the official metric used in HASOC 2023. In contrast, for Gujarati, the results showed a remarkable improvement of 21.64%.

Table 3
System performance on HASOC 2023 test-data (in percent)

Task/Team	RUN	Macro Precision	Macro Recall	Macro F1-score	F1 Improvement to baseline (RUN1)
Task 1A (Sinhala)					
Proposed method	RUN1	71.20	67.49	68.73	-
	RUN2	75.34	73.13	73.69	4.96
	RUN3	73.93	74.55	**74.10**	5.37
FiRC-NLP	First Rank	83.82	83.68	**84.00**	
LEGEND	Last Rank	55.88	55.72	55.75	
Task 1B (Gujarati)					
Proposed method	RUN1	34.28	50.00	40.67	-
	RUN2	62.22	63.69	62.31	21.64
	RUN3	69.30	72.19	**66.75**	26.08
FiRC-NLP	First Rank	83.92	86.38	**84.88**	
Gradient Descenders	Last Rank	67.12	66.26	66.62	

The oversampling process applied to the Sinhala language resulted in performance improvements compared to the imbalanced dataset, with an increase of 0.97% compared to RUN2, and a substantial 5.37% improvement when compared to RUN1. On the small balanced Gujarati data, treating emojis as single tokens boosted the F1-score performance by 4.44% compared to RUN2. Meanwhile, for RUN1 which use bag of words feature, there was a remarkable improvement of 26.08%. This enhancement can be attributed to emojis' presence within tweet contexts, where they influence the formation of word embedding vectors. This enriches the language model's comprehension of emotional and contextual cues within the text, thereby improving classification accuracy.

4. Conclusion and Future Work

Our participation in HASOC 2023 proposed a strategy to improve the performance of machine learning, i.e. SVM classifier by implementing optimization and feature selection. Our experiments on the HASOC datasets shows that applying word embeddings as input features for SVM can improve the F1-score significantly, compared to the use of Bag of word vectors. By dimensionality reduced, the computation become more efficient due to omiting sparsity of the vectors.

Other optimizations are also have significant effect in improving the classification results. Evaluation on the Gujarati data-test shows that for a very small data train, treating emoji as special token in the word embeddings vector space can improve the F1 score more than 4%. For the Sinhala dataset, since the training data is not contain any emoji, this optimization does not works.

We are inspired to the problem introduce in HASOC 2023, specifically the available of traning data which is very small. We courious to implement this strategy to other language we understand well, and want to proof that this optimization strategy will work well. Our future work will investigate it to English and Bahasa Indonesia dataset, hoping that the result would be improve significantly compared to the baseline.

5. References

[1] T. De Smedt, S. Jaki, E. Kotzé, L. Saoud, M. Gwóźdź, G. De Pauw, and W. Daelemans, Multilingual Cross-domain Perspectives on Online Hate Speech, *CLiPS Technical Report Series*, vol. 8, pp. 1–24, Sep. 2018, *arXiv*:1809.03944.

[2] D. D. Luxton, J. D. June, and J. M. Fairall, Social Media and Suicide: A Public Health Perspective, *American Journal of Public Health*, vol. 102, no. Suppl 2, pp. S195–S200, May 2012, doi:10.2105/AJPH.2011.300608.

[3] P. Fortuna, J. Rocha Da Silva, J. Soler-Company, L. Wanner, and S. Nunes, A Hierarchically-Labeled Portuguese Hate Speech Dataset, in: *Proceedings of the Third Workshop on Abusive Language Online*, Florence, Italy, 2019, pp. 94-104.

[4] E. del Valle and L. de la Fuente, Sentiment analysis methods for politics and hate speech contents in Spanish language: a systematic review, *IEEE Latin America Transactions*, vol. 21, no. 3, pp. 408–418, Mar. 2023, doi:10.1109/TLA.2023.10068844.

[5] H. Mubarak, A. Rashed, K. Darwish, Y. Samih, and A. Abdelali, Arabic Offensive Language on Twitter: Analysis and Experiments, in: *WANLP 2021 - 6th Arabic Natural Language Processing Workshop,* Association for Computational Linguistics (ACL), 2021, pp. 126–135.

[6] X.-S. Vu, T. Vu, M.-V. Tran, T. Le-Cong, and H. T. M. Nguyen, HSD Shared Task in VLSP Campaign 2019:Hate Speech Detection for Social Good, *arXiv preprint*, arXiv:2007.06493 (2020).

[7] C. Bosco, F. Dell'orletta, F. Poletto, M. Sanguinetti, and M. Tesconi, Overview of the EVALITA 2018 Hate Speech Detection Task. In: *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, Turin, Italy, 2018.

[8] M. O. Ibrohim and I. Budi, "Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter, in: *Proceedings of the Third Workshop on Abusive Language Online,* Florence, Italy, 2019, pp. 46-57.

[9] T. Mandl, S. Modha, A. Kumar M, and B. R. Chakravarthi, Overview of the HASOC Track at FIRE 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam, Hindi, English and German, in: *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*, December 2020, pp. 29–32. doi: 10.1145/3441501.3441517.

[10] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. R. Pardo, P. Rosso, and M. Sanguinetti, SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter, in: *Proceedings of the 13th International Workshop on Semantic Evaluation*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 54–63. doi: 10.18653/v1/S19-2007.

[11] T. Mandl *et al.*, Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages, in: *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*, December 2019, pp. 14–17 doi:10.1145/3368567.3368584.

[12] T. Ranasinghe, K. North, D. Premasiri, and M. Zampieri, Overview of the HASOC Subtrack at FIRE 2022: Offensive Language Identification in Marathi, *arXiv preprint*, arXiv:2211.10163 (2022).

[13] S. Satapara, H. Madhu, T. Ranasinghe, A.E. Dmonte, M. Zampieri, P. Pandya, N. Shah, M. Sandip, P. Majumder, and T. Mandl, Overview of the HASOC Subtrack at FIRE 2023: Hate-Speech Identification in Sinhala and Gujarati, in: *Working Notes of FIRE 2023 - Forum for Information Retrieval Evaluation*, Goa, India, Dec 15-18, 2023.

[14] H. M. S. T. Sandaruwan, S. A. S. Lorensuhewa, and M. A. L. Kalyani, Sinhala Hate Speech Detection in Social Media using Text Mining and Machine learning, in: *19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, IEEE, Sep. 2019, pp. 1–8. doi: 10.1109/ICTer48817.2019.9023655.

[15] S. W. A. M. D. Samarasinghe, R. G. N. Meegama, and M. Punchimudiyanse, Machine Learning Approach for the Detection of Hate Speech in Sinhala Unicode Text, in: *20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, IEEE, Nov. 2020, pp. 65–70. doi: 10.1109/ICTer51097.2020.9325493.

[16] L. Gohil and D. Patel, A sentiment analysis of Gujarati text using Gujarati senti word net, *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9, pp. 2290–2292, Jul. 2019, doi: 10.35940/ijitee.i8443.078919.

[17] S. MacAvaney, H. R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, Hate speech detection: Challenges and solutions, *PLoS One*, vol. 14, no. 8, Aug. 2019, doi: 10.1371/journal.pone.0221152.

[18] G. Kovács, P. Alonso, and R. Saini, Challenges of Hate Speech Detection in Social Media: Data Scarcity, and Leveraging External Resources, *SN Comput Sci*, vol. 2, no. 2, Apr. 2021, doi: 10.1007/s42979-021-00457-3.

[19] A. Zikri and S. Agustian, Penerapan Support Vector Machine dan FastText untuk Mendeteksi Hate Speech dan Abusive pada Twitter, *Jurnal Media Informatika Budidarma*, vol. 7, no. 1, pp. 436–443, 2023, doi: 10.30865/mib.v7i1.5408.

[20] T. Ranasinghe, I. Anuradha, D. Premasiri, K. Silva, H. Hettiarachchi, L. Uyangodage, and M. Zampieri, Sold: Sinhala offensive language dataset, *arXiv preprint,* arXiv:2212.00851, (2022).

[21] T. Joachims, Text Classification, in: *Learning to Classify Text Using Support Vector Machines*, Boston, MA: Springer US, 2002, pp. 7–33. doi: 10.1007/978-1-4615-0907-3_2.

[22] R. Awad Mariette and Khanna, Support Vector Machines for Classification, in: *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, Berkeley, CA: Apress, 2015, pp. 39–66. doi: 10.1007/978-1-4302-5990-9_3.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient Estimation of Word Representations in Vector Space, *arXiv preprint*, arXiv:1301.3781 (2013).

[24] J. Pennington, R. Socher, and C. Manning, Glove: Global Vectors for Word Representation, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[25] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, Bag of Tricks for Efficient Text Classification, *arXiv preprint*, arXiv1607.01759 (2016).

[26] S. Agustian, R. Saputra, and A. Fadhilah, 'Feature Selection' with Pretrained-BERT for Hate Speech and Offensive Content Identification in English and Hindi Languages, in: *Forum for Information Retrieval Evaluation*, India, 2021.