2022 International Conference on ICT for Emerging Regions (ICTer): 166 - 171

Sinhala Hate Speech Detection in Social Media Using Machine Learning and Deep Learning

W.S.S. Fernando
Department of Statistics
Faculty of Science, University of Colombo,
Colombo, Sri Lanka
siumisandu@gmail.com

Dr. Ruvan Weerasinghe
University of Colombo School of Computing
35, Reid Avenue,
Colombo, Sri Lanka
arw@ucsc.cmb.ac.lk

Mr. E.R.A.D. Bandara
Department of Statistics
Faculty of Science, University of Colombo,
Colombo, Sri Lanka
anjana@stat.cmb.ac.lk

2022 22nd International Conference on Advances in ICT for Emerging Regions (ICTer) | 979-8-3503-4613-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICTer58063.2022.10024082

***Abstract*— Communication and presentation of beliefs became easier than in previous decades due to the rapid rise of information technology and computer science. Because social media is accessible worldwide via the internet, anyone can simply target someone or a group who adheres to a different culture or belief. While everyone has the freedom to express their own opinions, it should not be destructive, and everyone has the right to be free of hate speech. Because there are no automatic mechanisms for detecting hate speech on social media, anyone can be readily targeted. Because social media service providers do not have extensive linguistic expertise of some languages, such as Sinhala, it may take a few days for them to delete hate-related comments from the material after they become aware of them. As a result, detecting hate speech in the Sinhala language is an urgent and crucial task. Machine learning and deep learning based algorithms were employed in this study to automatically recognize Sinhala hate speeches broadcast on social media. Bag of words, Tf-idf, Word2Vec, and FastText feature extraction methods were used to extract features from the comments. Logistic Regression, Multinomial Naïve Bayes, Support Vector Machine, XGBoost, Random Forest machine learning models and CNN, RNN, LSTM deep learning models were trained using two pre-collected datasets with different sizes. The best six models were then chosen and test set performances were shown. According to this study, FastText with RNN has the greatest AUC ROC 0.71 with 70% accuracy for the test set.**

Keywords—Hate speech detection, Sinhala, Machine Learning, Deep Learning, Natural Language Processing

I. INTRODUCTION

In today's world, internet-based communication media has surpassed all others as the most widely used information communication technology. People may openly share their ideas, opinions, and views via this communication medium known as social media. This platform is also useful for communicating with friends and relatives, learning new things, sharing information, and being entertained. Facebook and other social networking sites serve as an online directory that allows users to find their friends, relatives, and colleagues by searching for them on social networking sites [1].

There are 4.2 billion active global social media users worldwide, 4.15 billion active global mobile social media users worldwide. In 2020, Facebook has been the most popular social network by global audience size [2].

Social media has taken the place of peoples' daily activities. In addition, today's social media is frequently used to promote hateful or violent messages, comments, or speeches. Online hate speeches are defined as any communication that disparages a person or a group basis on traits such as race, colour, ethnicity, gender, sexual orientation, nation, faith, or political views [3]. The primary concerns that arise with hate speech in social media are psychological damage and self-hatred, which encourage racist and sexist attitudes that may escalate to violence, hate speech silences women and minorities, and creates a disordered society [4]. Because of these concerns, identifying hate speech on social media is a significant topic of study.

Facebook eliminated 22.3 million pieces of hate speech content during the most recent reporting period, down from 31.5 million in the second quarter of 2021 [5]. In January 2021, there were 7.9 million social media users in Sri Lanka [6]. In January 2021, the overall popularity of social media users in Sri Lanka was 37% of the entire population. In October 2021, 71% of social media users were on Facebook, and 14% were on YouTube [7]. Facebook is the world's second most popular website and the most popular in Sri Lanka. Facebook could be identified as a special source of addiction. Worldwide, it is estimated that 210 million individuals are addicted to the internet and social media.

Many incidents of hate speech through social media, which might incite acts of violence, have been observed by social scientists and others [8]. During the March 2018 riots, the authorities shut down Facebook, WhatsApp, and Viber for a week, starting that Facebook had failed to respond appropriately in an emergency. Disinformation and hate speech circulated on Facebook in 2018 and 2019 after the Easter bombings, in Kandy and Negombo sparked an anti-Muslim rioting [9]. Hence, the rapid spread of race hate speech on social media appears to have a substantial influence on society and a country's goodwill.

Facebook users are obsessive about the rules, and their content is being hijacked by tens of thousands of contractors accused of mediating. Activists and journalists have been banned in a number of nations and disputed territories. Some social media corporations (Facebook, Google, and Twitter) are becoming quicker at responding to hate speech online in order to combat these damaging hate reactions. During 2018, these internet giants erased 72% of illegal hate speech from their platforms [10].

Problems develop when social media platforms artificial intelligence is weakly tailored to local languages, and enterprises devote a limited amount of money in fluently staffing them. This was especially serious in Myanmar, where Reuters

979-8-3503-4613-8/22//$31.00 ©2022 IEEE

claimed that Facebook only employed two Burmese speakers by early 2015. Experts warned of a ripe atmosphere on Facebook for propagating hate speech after a series of anti-Muslim acts of violence began in 2012. Since the posts and comments in Sri Lanka were in Sinhala or Romanized Sinhala, existing technologies failed to identify racial statements and offensive language. The manual process of identifying and removing hate speech content takes lot of time and effort.

All of these point to the necessity of recognizing hate speech and deleting it from social media. This study develops and assesses machine learning and deep learning based methods to easily identify Sinhala hate speech in social media.

II. Related Works

Dias et al. [11] have conducted a racist speech detection study in Sinhala. They have used racist comments posted on social media using Sinhala Unicode. In their study, they have employed a corpus of 73 racism observations and 111 non-racist observations retrieved from social media. They have utilized the SVM classification model with the word n-gram feature extraction method and achieved a precision of 100% and an accuracy of 70.8%. This research has used very few comments and has not used a balanced dataset for the model training. Also, they have used only one feature extraction method and one classification method.

In the research conducted by Hettiarachchi et al. [12], they have employed a model to identify hate speech written by the Romanized Sinhala language. According to their research, they have used several feature extraction methods and classification techniques as Logistic Regression, MNB, SVM, and Random Forest. They have obtained 70.4% test accuracy using the MNB classifier with the TF-IDF feature extraction method.

Sandaruwan et al. [13] have been done Sinhala hate speech detection research with the balanced dataset containing 1000 comments per three categories: hate comments, offensive comments, and neutral comments. They have conducted five experiments to create a good model. The 300 comments from the above dataset were utilized in all five studies, and the datasets were balanced between three classes. They have employed five experiment. According to their research, the MNB classifier performed well with every group, and character trigram features have performed better than other feature types.

In Abro et al.'s research [14], they have employed three different feature extraction strategies as well as eight different machine learning algorithms. Bigram features using the support vector machine algorithm outperformed the others, with maximum accuracy of 79%. They have used 2902 tweets, and among them, 16% hate speeches, 50% of the speeches were not offensive, and 33% were offensive but not hateful. Since this dataset is imbalanced, it would be better to use either undersampling or oversampling techniques to increase model accuracy.

Mozafari et al. [15] have used a transfer learning strategy founded on a pre-trained language model termed BERT in their study. They have specifically looked into BERT's ability to capture hateful content inside social media content utilizing novel fine-tuning methods based on transfer learning. They used two widely accessible datasets that were marked up for racism, sexism, hate, or offensive content on Twitter to evaluate the suggested approach. They have obtained the best results as of F1 score of 88% and 92% for two datasets when using CNN with pre-trained BERT model.

Most of the studies that found on hate speech detection using deep learning have been done for the English language. Some other languages have been considered. Kamble & Joshi [16] have used CNN-1D, LSTM, and BiLSTM to experiment with a benchmark dataset of English-Hindi code-mixed tweets. The researchers then discovered that utilizing domain-specific embeddings improves target group representation. They also demonstrated that their models improve the F1-score by roughly 12% when compared to previous work that used statistical classifiers. Alshalan & Al-Khalifa [17] have employed a deep learning strategy for automatic hate speech detection in Arabic tweets. They have tested CNN, GRU, CNN+GRU, and BERT models on two datasets as part of their research. With an F1-score of 0.79 and AUC ROC of 0.89, they discovered that the CNN model produced the best results.

III. Methodology

The proposed research process illustrates the functioning flow of the process stages and stands as the blueprint of how the intended model is developed step by step.

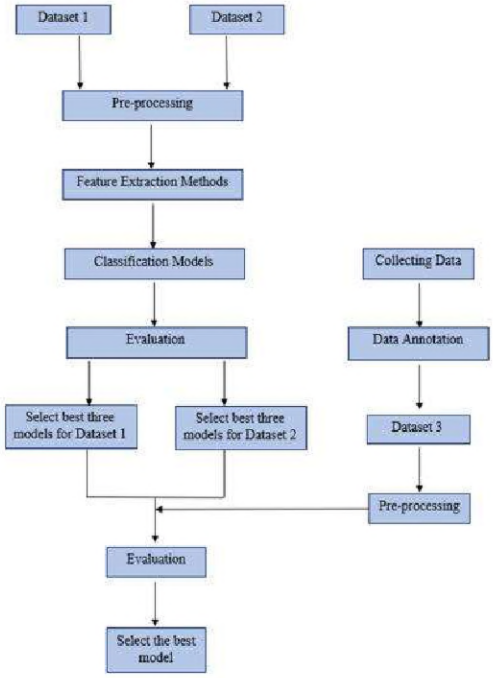

Fig. 1. Research Process

A. Data sets used in this study

This study utilizes three datasets. Datasets gathered from social media such as Facebook, Twitter, and YouTube, have been used to train a model to detect Sinhala hate speech in social media. Dataset 1 and 2 were pre-collected datasets with different sizes and used two datasets for training models. Those datasets were manually labeled as hate speech or not.

It is essential to test model performance using a completely different dataset. Therefore, Facebook comments were collected using the Instant Data Scraper tool and manually annotated observations as hate and neutral by a majority vote of 3 independent raters. That dataset was taken as Dataset 3.

TABLE I. DATASETS DESCRIPTION

Dataset	*Hate Speeches*	*Neutral Speeches*	*Total*
Dataset 1	793(45.5%)	949(54.5%)	1742
Dataset 2	3455(54.5%)	2890(45.5%)	6345
Dataset 3	137(34.2%)	263(65.8%)	400

B. Data Pre-processing

Text pre-processing is one of the main steps in natural language processing tasks. It converts text into an easier-to-understand format so that machine learning algorithms can perform much better.

Fig. 2. Data Preprocessing steps

First, removed non-Sinhala characters since Sinhala Unicode posts were used to train the models. Then under the remove special characters, punctuation marks and numbers were removed. Then tokenized the comments, and word frequencies were observed in descending order.

Afterward, removed stop words that do not influence the comment being hate or not. The words like "මේ", "එක", "වගේ" belongs to the stop words. By removing stop words, the dimensionality of the feature vector can be reduced. Publically available stop word list [18] was used. Here, the most commonly occurring terms were chosen, and certain words were removed from the list because they changed the polarity of a phrase.

TABLE II. REMOVE WORDS FROM STOP WORD LIST

Removed words from Sinhala stop word list		
නොමැති	නැත	නැහැ
යහපත්	නෑ	නොවේ
බැහැ	නැති	නොව
බැරි	හොඳ	නොහැකි
හොඳයි	එපා	විනාශ

Finally, the stemming part was done using pre-collected Sinhala stemming dictionary. In Sinhala stemming dictionary, stems were listed in alphabetical order. Stemming is the process of eliminating prefixes and suffixes from words in order to reduce them to basic words. Stemming also reduces the dimensionality of the feature vector [13]. Table III shows a few stem roots and words in Sinhala stemming dictionary.

Words	*Stem*
අංශයකට, අංශයකම, අංශයක්	අංශය
අකමැත්තක්, අකමැත්තකින්	අකමැත්ත
ඉටුවීම, ඉටුවීමක්, ඉටුවීමට	ඉටුවී
ජාතිකයන්ගෙන්, ජාතිකයන්ගේ	ජාතිකයන්

TABLE III. STEM VARIATIONS AND STEM

C. Feature Extraction Methods

Text classification relies heavily on feature extraction, which has a direct impact on text classification accuracy [19]. Feature extraction is the technique of obtaining a selection of key characteristics from text input in order to improve classification performance. Machine learning algorithms are unable to deal with unprocessed text. Hence, several feature extraction methods are required.

1) Bag-of-words

A bag-of-words is a text representation that describes the frequency with which words appear in a document. BOW ignores the grammar and word order but records the occurrence frequency of each word with the word. This is the simplest way to represent text in numbers. Using BOW, a sentence can be represented as a bag of words vector.

2) TF-IDF

Term Frequency – Inverse Document Frequency is abbreviated as TF-IDF. The TF-IDF is a statistic that not only measures the frequency of a word in the corpus, but also its relevance. Higher values of Tf-idf signify higher importance of the words in the corpus, while lower values represent lower importance.

D. Word Embeddings

All texts must be presented as a vector of numbers in neural networks and machine learning. Same as the feature extraction methods, word embeddings turn a word into a vector that encapsulates the semantics behind it. In particular word embeddings bring related words closer together in the representation space. As a result, utilizing word embeddings as input to neural models helps them to generalize beyond the unique words contained in the input sentence or documents while assigning the sentiment class [20].

1) FastText

FastText is a library developed by Facebook's AI Research (FAIR) team for fast learning of word embeddings and text classification. There are two ways to use FastText word embeddings. One way is to use pre-trained word vectors for the Sinhala language and the other way is to create word vectors for the training dataset by using fastText. In this study, pre-trained word vectors in fastText have been used for the Sinhala language.

2) Word2Vec

Word2Vec is a standard method of generating word embeddings in a corpus. Word2Vec builds vectors with dispersed numerical representations of word properties, such as the context of individual words, by grouping the vectors of related words together in vector space. Based on post appearances, it can make very accurate assumptions about meaning of a word.

Gensim is a free Python library for natural language processing that includes a Word2Vec class for interacting with Word2Vec models. There are two different types of training

algorithms. CBOW (Continuous Bag of Words) and skip grams are their names.

E. Classifiers

Classification is a data-mining technique that classifies a collection of data into categories to aid prediction and analysis. For a given one or more inputs, classification models will attempt to predict the value of one or more outcomes. In this research, the outcomes are hate (1) or neutral (0) labels. Here, machine learning and deep learning models are employed to detect Sinhala hate speech on social media.

Logistic Regression, Support Vector Machine, Multinomial Naïve Bayes, Random Forest, and XGBoost machine learning modes were used. As deep learning models CNN, RNN, and LSTM were used.

F. Hyper-parameter Tuning

Both training datasets were used to fit all machine learning and deep learning models. The results of each model are presented in the following section. Following those results, three models from each dataset were chosen as the best models. For selected machine learning models, hyper-parameter tuning was done by using Grid Search in scikit-learn.

Then the KerasTuner was used for deep learning model hyper-parameter tuning. KerasTuner is a flexible, convenient hyper-parameter optimization toolkit for deep learning models which is used to identify the best hyperparameter values. Bayesian Optimization, Hyperband, and Random Search methods are all included in KerasTuner toolkit.

IV. Experimental Results

Machine learning model results for word unigram, bigram, and trigram as well as character unigram, bigram, and trigram were observed under TfidfVectorizer and CountVectorizer. Also, observed results for Skip-grams and CBOW under Word2Vec. The outcomes of the experiment covered in this section.

A. Observed Results for Dataset1

The Dataset 1 results are separated and explained below using the feature extraction method.

The character trigram feature in CountVectorizer fared better than the other n-grams when compared to the character n-grams. While employing the character and word n-grams features from CountVectorizer, the MNB classifier was effective at identifying hate speech, and the character n-grams features performed well on Dataset 1.

The character n-grams feature performed well on Dataset 1 when employing the TfidfVectorizer's word and character n-grams features, which work similarly to the CountVectorizer.

The following section discusses how machine learning models with word2Vec skip-grams and CBOW performed. The MNB and XGBoost classifiers are unsuccessful with negative values. Therefore, only the LR, SVM, and RF models were taken into account while using Word2Vec.

The observed results show that the Word2Vec CBOW-based RF classifier achieved 70% accuracy and a 0.62 F1-Score. In comparison to Skip-grams, Word2Vec CBOW fared better with machine learning models.

The MNB classifier with CountVectorizer character trigram outperformed the other models in terms of machine learning model performance on the dataset 1. Character n-grams outperformed word n-grams in terms of performance.

TABLE IV. Deep Learning Models for Dataset 1

Classifier	*Deep Learning models*			
	Without FastText		*With FastText*	
	Accuracy	*F1-score*	*Accuracy*	*F1-score*
CNN	0.72	0.63	0.74	0.69
RNN	0.75	0.73	**0.76**	**0.69**
LSTM	0.69	0.68	0.73	0.69

When compared the performance of deep learning models with and without FastText word embeddings, deep learning models with FastText outperformed those without. With 76% accuracy and a 0.69 F1-Score, the RNN model outperformed the other models.

B. Observed Results for Dataset 2

The Dataset 2 results are divided and discussed below using the feature extraction method, the.

Word unigram performed well on Dataset 2 when compared to the word n-grams formed by CountVectorizer. In the character trigram formed by CountVectorizer, LR classifier functioned excellently. It provided an F1-score of 0.87 and an accuracy of 86%. When compared to the word n-grams and character n-grams formed by the CountVectorizer, character trigram performed well on Dataset 2.

According to the TfidfVectorizer word trigram results, the LR, SVM, MNB, and RF classifiers performed well. The word unigram feature in TfidfVectorizer fared better than the other n-grams when compared to the word n-grams. Comparing the TfidfVectorizer word n-grams and character n-grams to character trigram, character trigram performed well on Dataset 2.

Based on the data, the RF classifier employing Word2Vec CBOW achieved 78% accuracy and a 0.80 F1-Score. Word2Vec CBOW outperformed Skip-grams in machine learning models on the Dataset 2.

The LR and SVM classifiers using the TfidfVectorizer character trigram outperformed the other machine learning models. In this instance, character n-grams conquered over word n-grams as well.

TABLE V. Deep Learning Models for Dataset 2

Classifier	*Deep Learning models*			
	Without FastText		*With FastText*	
	Accuracy	*F1-Score*	*Accuracy*	*F1-Score*
CNN	0.86	0.87	0.86	0.88
RNN	0.85	0.86	**0.88**	**0.89**
LSTM	0.83	0.84	0.86	0.87

CNN, RNN, and LSTM models with FastTest fared better in this dataset as well, with RNN providing 88% accuracy and 0.89 F1-Score.

C. Summary of selected best models

Among the fitted models, six best fit models were chosen. These six models contain the three best models from dataset1 and three from dataset 2. The summary of the selected models is as follows.

1) Dataset 1

TABLE VI. SELECTED BEST MODELS FOR DATASET 1

Classifier	*Accuracy*	*F1 Score*
Model 1: MNB (CountVectorizer character trigram)	0.79	0.78
Model 2: SVM (TF-IDF character trigram)	0.79	0.76
Model 3: RF (TF-IDF character trigram)	0.79	0.76

2) Dataset 2

TABLE VII. SELECTED BEST MODELS FOR DATASET 2

Classifier	*Accuracy*	*F1 Score*
Model 4: LR (TF-IDF character trigram)	0.87	0.88
Model 5: SVM (TF-IDF character trigram)	0.87	0.88
Model 6: RNN (FastText)	0.88	0.89

D. Summary of the best model performances on the Test set

TABLE VIII. BEST MODEL PERFORMANCES ON THE TEST SET (DATASET 3)

Classifier	*Accuracy*	*F1 Score*	*Accuracy*	*AUC ROC*
Model 1	Dataset 1	0.59	0.52	0.66
Model 2	Dataset 1	0.68	0.41	0.68
Model 3	Dataset 1	0.69	0.38	0.68
Model 4	Dataset 2	0.70	0.53	0.71
Model 5	Dataset 2	0.70	0.53	0.71
Model 6	**Dataset 2**	**0.70**	**0.58**	**0.71**

Models trained with Dataset 2 (dataset which contains a large number of observations) performed better than models trained with Dataset 1(dataset which contains a small number of observations). On the testing dataset, the RNN with FastText classifier outperformed the other models and the LR and SVM classifiers with TF-IDF feature extraction performed equally well.

E. Selected best model

	precision	recall	f1-score	support
0	0.78	0.75	0.77	260
1	0.56	0.61	0.58	137
accuracy			0.70	397
macro avg	0.67	0.68	0.68	397
weighted avg	0.71	0.70	0.71	397

Fig. 3. Classification report for RNN model

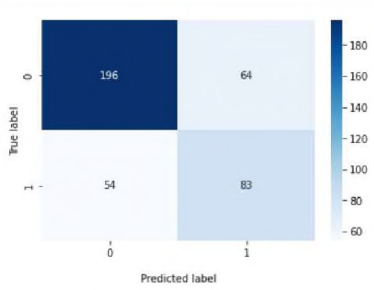

Fig. 4. Confusion matrix for RNN model

RNN deep learning model gives 0.70 accuracy and 0.58 F1-score. Here 118 comments were misclassified. 64 neutral speeches were misclassified as hate speeches and 54 hate speeches were misclassified as neutral speeches.

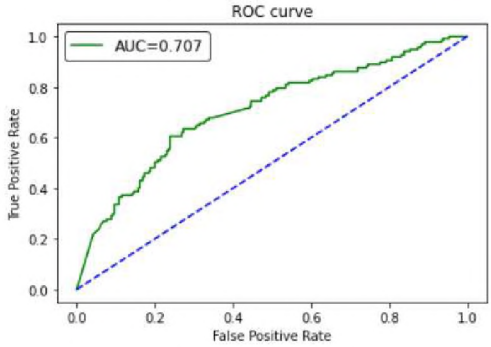

Fig. 5. ROC curve for RNN model

This classifier had an AUC ROC score of 0.707 when the dataset 2 was used to train with FastText word embeddings.

V. LIMITATIONS

Main goal of this study was to identify hate speech, no further study or machine learning was done to determine which community, sub-community, or group of people a particular hate speech targeted. Social media is a platform where people from various social, religious, and cultural backgrounds may interact and communicate with one another. As a result, identifying the target group for hate speech content will be critical information for the hate speech detection procedure, and the lack of this feature can be regarded as a limitation of this project.

VI. CONCLUSION

The best features and classification models that can be employed in Sinhala hate speech recognition with machine learning and deep learning have been identified and introduced in this work. Two pre-collected datasets were used to train the machine learning and deep learning models in this study. There are 1742 observations in Dataset 1 and 6345 observations in Dataset 2. Five different machine learning models and three different deep learning models were assessed and compared using these datasets. Machine learning models include Logistic

Regression, Support Vector Machine, Multinomial Naive Bayes, Random Forest, and XGBoost, while deep learning models include CNN, RNN, and LSTM. In order to generate the optimal model, different feature extraction and word embeddings were used. There are 400 observations in the test dataset, which are manually annotated as hate or neutral. The best six models' performance was attained using this dataset. With an accuracy of 0.70 and an AUCROC of 0.71, the results showed that RNN with FastText embeddings beats other models. There are some researches also got around 0.70 accuracy for detecting Sinhala hate speech [12], [13].

Character trigram features surpass all other feature types in machine learning models. Because many hate speeches and offensive speeches are posted with spelling mistakes and replaced with comparable characters, character n-gram features work well in hate speech recognition. Deep learning models perform better when FastText embeddings are used instead of no embedding.

When comparing machine learning and deep learning models for detecting hate speech in Sinhala, deep learning models outperform machine learning models. In order to obtain the optimum performance measurements, it is necessary to tune the hyperparameters in both machine learning and deep learning.

VII. FUTURE WORKS

This study focused on supervised learning and it is possible to use unsupervised learning techniques to identify Sinhala hate speeches.

According to literature, the stemming and stop word removal techniques in the pre-processing steps do not significantly effect the accuracy of sentiment analysis in Indonesian text documents . Thus, future research into this aspect of the Sinhala language can be carried out.

In deep learning models, more hidden layers could be employed in the models used in this work. More bidirectional layers can be added to an RNN model is an example. Transfer learning techniques can be utilized using transformers such as BERT and GPT models. So, those model performances on Sinhala hate speech detection may also be carried out.

REFERENCES

[1] Kirschner, P. A., & Karpinski, A. C. (2010). Facebook® and academic performance. Computers in Human Behavior, 26(6), 1237–1245. https://doi.org/10.1016/j.chb.2010.03.024

[2] Statista. (2021, January). Internet users in the world 2021. https://www.statista.com/statistics/617136/digital-population-worldwide/

[3] Zhang, Z., & Luo, L. (2019). Hate speech detection: A solved problem? The challenging case of long tail on Twitter. Semantic Web, 10(5), 925–945. https://doi.org/10.3233/SW-180338

[4] Ring, C. (2013). Hate Speech IN Social Media: An Exploration of The Problem And Its Proposed Solutions. Journal of Chemical Information and Modeling, 53(9), 1689–1699.

[5] Statista. (2021, November). Facebook hate speech removal per quarter 2021. https://www.statista.com/statistics/1013804/facebook-hate-speech-content-deletion-quarter/

[6] Kemp, S. (2021, February 12). Digital in Sri Lanka: All the Statistics You Need in 2021. DataReportal, Global Digital Insights. https://datareportal.com/reports/digital-2021-sri-lanka

[7] Statcounter (2021, October). Social Media Stats Worldwide. https://gs.statcounter.com/social-media-stats

[8] Laub, Z. (2019, June 7). Hate Speech on Social Media: Global Comparisons. Council on Foreign Relations (cfr.org). https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons

[9] Amarasingam, A., & Rizwie, R. (2021, March 5). Turning the Tap Off: The Impacts of Social Media Shutdown After Sri Lanka's Easter Attacks. GNET (gnet-research.org) https://gnet-research.org/2021/03/05/turning-the-tap-off-the-impacts-of-social-media-shutdown-after-sri-lankas-easter-attacks/

[10] European Commission. (2019). Code of Conduct on countering illegal hate speech online. Fourth Evaluation. February. https://ec.europa.eu/commission/presscorner/detail/en/IP_19_805

[11] Dias, D. S., Welikala, M. D., & Dias, N. G. J. (2019). Identifying Racist Social Media Comments in Sinhala Language Using Text Analytics Models with Machine Learning. February 2019, 1–6. https://doi.org/10.1109/icter.2018.8615492

[12] Hettiarachchi, N., Weerasinghe, R., & Pushpanda, R. (2020). Detecting hate speech in social media articles in romanized sinhala. 20th International Conference on Advances in ICT for Emerging Regions, ICTer 2020 - Proceedings, January 2021, 250–255. https://doi.org/10.1109/ICTer51097.2020.9325465

[13] Sandaruwan, H. M. S. T., Lorensuhewa, S. A. S., & Kalyani, M. A. L. (2019). Sinhala Hate Speech Detection in Social Media using Text Mining and Machine learning. 19th International Conference on Advances in ICT for Emerging Regions, ICTer 2019 - Proceedings, 250, 1–8. https://doi.org/10.1109/ICTer48817.2019.9023655

[14] Abro, S., Shaikh, S., Ali, Z., Khan, S., Mujtaba, G., & Khand, Z. H. (2020). Automatic hate speech detection using machine learning: A comparative study. International Journal of Advanced Computer Science and Applications, 11(8), 484–491. https://doi.org/10.14569/IJACSA.2020.0110861

[15] Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. Studies in Computational Intelligence, 881 SCI, 928–940. https://doi.org/10.1007/978-3-030-36687-2_77

[16] Kamble, S., & Joshi, A. (2018). Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models. http://arxiv.org/abs/1811.05145

[17] Alshalan, R., & Al-Khalifa, H. (2020). A deep learning approach for automatic hate speech detection in the saudi twittersphere. Applied Sciences (Switzerland), 10(23), 1–16. https://doi.org/10.3390/app10238614

[18] Language Technology Research Laboratory [Online] Available: http://ltrl.ucsc.lk/

[19] Singh, Vandita, Kumar, B., & Patnaik, T. (2013). Feature Extraction Techniques for Handwritten Text in Various Scripts : a Survey. 1, 238–241.

[20] Biswas, E., Vijay-Shanker, K., & Pollock, L. (2019). Exploring word embedding techniques to improve sentiment analysis of software engineering texts. IEEE International Working Conference on Mining Software Repositories, 2019-May, 68–78. https://doi.org/10.1109/MSR.2019.00020

[21] Pradana, A. W., & Hayaty, M. (2019). The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 4(3), 375–380. https://doi.org/10.22219/kinetik.v4i4.912