Identifying Hate Speech in Low Resource Language: Sinhala

Enduri Jahnavi (21BDS019)
Data Science and Artificial Intelligence
Indian Institute of Information Technology, Dharwad
Karnataka, India

***Abstract*—Sentiment analysis in low-resource languages poses unique challenges due to limited linguistic resources and labeled datasets. This paper focuses on Sinhala, a low-resource language predominantly spoken in Sri Lanka, emphasizing the importance of understanding sentiment for public opinion, social trends, and communication strategies. The study explores machine learning and deep learning models, including BERT, ensemble learning, and various deep learning architectures, to address the challenges of sentiment analysis in Sinhala. The effect of stacking the same and different Deep Learning model on top of each other and compare them based on accuracy and F1 score is studied. A combined CNN-LSTM-BERT model for detecting hate speech in Sinhala text is proposed. Results showcase that among the machine learning models, Logistic Regression outperformed others, achieving the highest accuracy at 68.40%. In the realm of deep learning models, the combination of BERT, CNN, and LSTM stood out with the highest accuracy of 80.07%. The study also highlights the significance of thoughtful model selection and parameter tuning for optimal sentiment classification in low-resource languages through various experiments on model architectures and varying hyperparameters.**

***Index Terms*—Hate Speech, Sinhala, NLP, Ensemble Models.**

I. Introduction

Sentiment analysis, a field within natural language processing, plays a pivotal role in deciphering the emotional tone and opinions expressed in textual content. While sentiment analysis has seen widespread application in high-resource languages, its significance becomes even more important in the context of low-resource languages. Among these languages, Sinhala, spoken predominantly in Sri Lanka, emerges as a particularly noteworthy case. Understanding sentiment in Sinhala text is crucial for various reasons, including gauging public opinion, monitoring social trends, and enhancing communication strategies in the digital era.

Low-resource languages, often lacking extensive linguistic resources and labelled datasets, present unique challenges for sentiment analysis. The scarcity of available tools and datasets makes it difficult to develop accurate and contextually relevant sentiment analysis models. Despite these challenges, the importance of sentiment analysis in low-resource languages cannot be overstated. It opens avenues for better understanding community sentiments, cultural nuances, and social dynamics, providing valuable insights for decision-making and policy formulation. [3]

Considerable efforts have been dedicated to exploring sentiment analysis in low-resource languages like Indonesian, Sinhala, Marathi, and Gujarati. Researchers have employed diverse strategies, including custom tokenization and lemmatization methods, or context-aware stopword removal, [9], [12] as well as machine learning and ensembles of machine learning algorithms [5], [8], [13], [17]. More recently, attention has shifted towards the implementation of deep learning models such as Long Short-Term Memory (LSTM) [19] and Bidirectional Encoder Representations from Transformers (BERT). BERT itself has seen various adaptations, including models like RoBERTa, FinBERT, SinhalaBERT, and MahaBERT, reflecting an ongoing commitment to refining sentiment analysis techniques for languages with limited linguistic resources [7], [10], [11], [14], [15]. These advancements showcase the versatility and adaptability of modern approaches, aiming to improve the accuracy and effectiveness of sentiment analysis across diverse linguistic landscapes.

This paper delves into the significance of sentiment analysis in low-resource languages, with a specific focus on the importance of understanding sentiment in Sinhala text. Through the exploration of various machine learning and deep learning models, aim is to address the challenges associated with sentiment analysis in Sinhala and contribute to the development of effective tools for this unique linguistic context. This work aims to answer two Research Questions:
- RQ1: How can we effectively forecast sentiment in Sinhala text using machine learning and deep learning techniques?
- RQ2: What impact does the alteration of established deep learning models through the stacking of similar and dissimilar models have on overall performance?

II. Background

In this section, A background encompassing the key concepts and techniques necessary for this work is explained. This primarily involves a comprehensive exploration of word embedding techniques and an overview of the various deep learning models that constitute integral components of the study.

A. Word Embeddings

1) Word2Vec: is a word embedding technique that is applied to convert words into dense vectors based on their

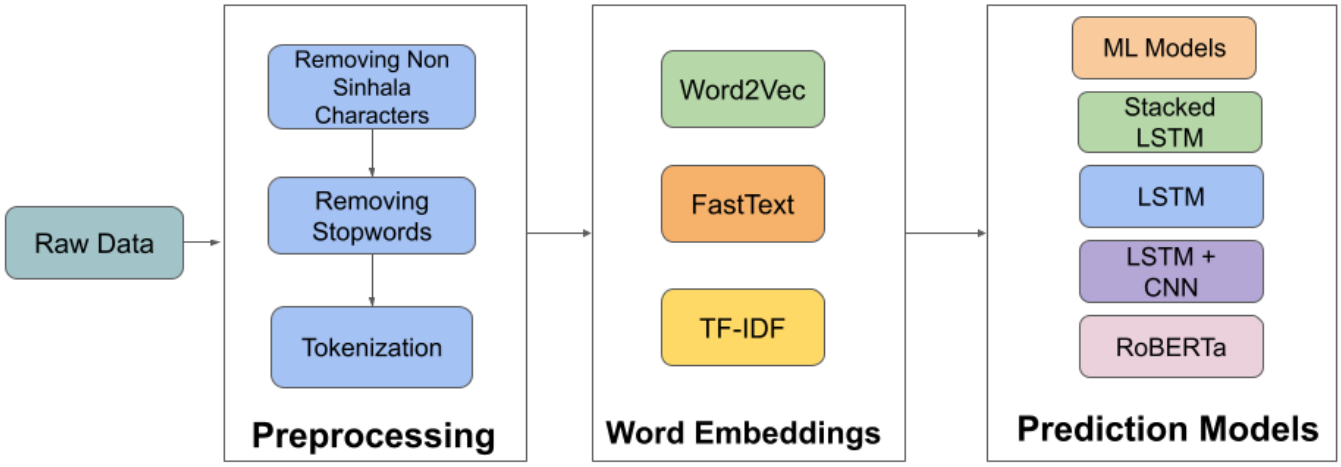

Fig. 1. WorkFlow

contextual usage. By leveraging the local context of words, Word2Vec generates embeddings that encapsulate semantic relationships, creating a better understanding of word meanings within the Sinhala language corpus.

2) TF-IDF: TF-IDF, which stands for Term Frequency-Inverse Document Frequency, is a numerical statistic widely used in natural language processing and information retrieval. It quantifies the importance of a term within a document relative to its occurrence in a collection of documents. The TF component measures how frequently a term appears in a specific document, while IDF gauges the rarity of the term across the entire document collection.

B. Ensemble learning

is a powerful machine learning technique that combines the predictions of multiple individual models to enhance overall performance and robustness. By aggregating the insights from diverse models, ensemble methods, such as Random Forest and XGBoost, aim to mitigate individual weaknesses and improve predictive accuracy. The accuracy achieved through ensemble learning leverages the collective intelligence of constituent models, often resulting in superior results compared to individual models. This approach proves particularly effective in addressing complex tasks, contributing to increased generalization and resilience against overfitting, making ensemble learning a widely adopted strategy across various machine learning applications.

C. Deep Learning Models

1) ANN (Artificial Neural Network): Artificial Neural Networks (ANNs) form the foundation of our deep learning approach. These networks, inspired by the human brain, help us uncover complex patterns within Sinhala text. By connecting nodes in a way that allows for non-linear mapping of input features, ANNs contribute to the model's ability to predict hate speech.

2) BERT: Incorporation of Bidirectional Encoder Representations from Transformers (BERT), a state-of-the-art transformer-based model, to enhance the understanding of Sinhala language nuances. BERT's strength lies in considering bidirectional context, making it good at grasping the subtleties within Sinhala text. This addition enriches the model's contextual awareness, particularly valuable for hate speech detection. [7].

3) LSTM: Long Short-Term Memory (LSTM) networks are employed to understand the order of words in Sinhala text. LSTMs address challenges related to long-range dependencies, crucial for recognizing contextual details in hate speech. Their contribution enhances the model's ability to capture sequential patterns effectively. [4], [6]

4) Stacked LSTM: The Stacked Long Short-Term Memory (LSTM) architecture is a more intricate variation that involves stacking multiple LSTM layers on top of each other. Each LSTM layer is responsible for capturing different levels of abstraction in the sequential data, with the output of one layer serving as the input for the subsequent layer. This stacking mechanism enables the model to learn hierarchical representations of sequential patterns, allowing for a more nuanced understanding of complex dependencies within the Sinhala language data [].

5) LSTM + CNN: The integration of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks results in a hybrid architecture designed to address the intricacies of hate speech prediction in Sinhala. CNNs are adept at capturing spatial hierarchies and recognizing patterns within local neighbourhoods of the input data. On the other hand, LSTMs specialize in modeling sequential dependencies and understanding patterns that evolve over time. By combining these architectures, the hybrid model aims to capitalize on the complementary strengths of CNNs and LSTMs. The CNN component focuses on spatial features and patterns in the Sinhala language, while the LSTM component delves into the

post_id int64 563,324… 1,488,9…	text string · *lengths* 18 634	tokens string · *lengths* 15 643	rationales string · *lengths* 2 369	label string · *classes* 2 values
726,758,237,668,659,200	@USER @USER පව්ට පට පට...	@USER @USER පව්ට පට පට . . .	[]	NOT
915,618,589,855,617,000	පරණ කැල්ල අද වෙනකම් හිටියනම් අදට අවුරුදු 4යි. යාලු වෙලා.…	පරණ කැල්ල අද වෙනකම් හිටියනම් අදට අවුරුදු 4යි .…	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…	OFF
925,001,070,430,040,000	යාළුවා කියලා හිතන් සර් ගේ ඔලුවට රෙද්ද දාලා නෙලන එක…	යාළුවා කියලා හිතන් සර් ගේ ඔලුවට රෙද්ද දාලා නෙලන එ…	[]	NOT
1,397,219,745,707,987,000	හොඳ මිතුරියක් කතා කලා. විස්තර කතාකරමින් ඉදලා මේ දවස්වල…	හොඳ මිතුරියක් කතා කලා . විස්තර කතාකරමින් ඉදලා මේ…	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…	OFF
950,376,113,150,222,300	ඔය බනින්නෙ.. හරකා, මී හරකා කිය කිය...	ඔය බනින්නෙ . . හරකා , මී හරකා කිය කිය . . .	[0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0]	OFF
659,668,698,542,702,600	කැනරින් මංගල හමුවෙයි: ඇමරිකානු රාජ්‍ය දෙපාර්තමේන්තු…	කැනරින් මංගල හමුවෙයි : ඇමරිකානු රාජ්‍ය…	[]	NOT
746,759,888,664,035,300	රූ. ඩැනියල් දවල් මිගෙල් is on Swarnawahini හම්මේ.. ගොන්…	රූ. ඩැනියල් දවල් මිගෙල් is on Swarnawahini හම්මේ . .…	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,…	OFF
715,019,979,393,863,700	@USER @USER අඩෝ.. බර්ත්ඩේ එක දවසේ පුෂ පලාගන්න ආසද :v	@USER @USER අඩෝ . . බර්ත්ඩේ එක දවසේ පු *…	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0]	OFF
1,314,493,794,851,713,000	නිරෝදායනයට යන්න එනවද කියලා මීනාක්ශිගෙන් ඇහුවා එන්න බෑලු …	නිරෝදායනයට යන්න එනවද කියලා මීනාක්ශිගෙන් ඇහුවා…	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,…	OFF
1,207,341,992,465,719,300	ඇත්ත කියනව ජෝං බාස්.... තමුසෙ නේද පිස්තෝලෙ කටට…	ඇත්ත කියනව ජෝං බාස් . . . . තමුසෙ නේද පිස්තෝලෙ…	[]	NOT
998,584,011,000,692,700	@USER @USER අරූට ඉතින් එලොව පොල් පෙනුනලු...ලැජ්ජාවට හිනා…	@USER @USER අරූට ඉතින් එලොව පොල් පෙනුනලු . . .…	[]	NOT
1,173,215,765,274,583,000	@USER පොහොට්ටුව පිපෙන්ඩයි යන්නෙ. @USER @USER @USER…	@USER පොහොට්ටුව පිපෙන්ඩයි යන්නෙ . @USER @USER @USE…	[]	NOT

Fig. 2. Snapshot of the Sinhala Offensive Language Dataset (SOLD)

temporal aspects, capturing the sequential nature of language evolution. This collaborative approach seeks to provide a more comprehensive understanding of complex relationships within the data, ultimately enhancing the model's ability to predict hate speech effectively in the Sinhala language context. [1], [2], [18]

6) BERT + CNN + LSTM : The BERT + CNN + LSTM model is a proposed architecture that combines the strengths of three powerful components:
- BERT (Bidirectional Encoder Representations from Transformers) contributes advanced language understanding,
- CNN (Convolutional Neural Network) captures spatial hierarchies, and
- LSTM (Long Short-Term Memory) models sequential dependencies.
This mix of Deep Learning models aims to harness the contextual understanding of BERT along with the spatial and sequential insights provided by CNN and LSTM, creating a comprehensive model for tasks like sentiment analysis in Sinhala text.

III. Methodology

In this section, an in-depth overview of the approach for sentiment classification of Sinhala text is provided.this study utilizes machine learning and deep learning techniques, employing algorithms like word2vec, fastText, and TF-IDF for word embedding. The model selection includes LSTM, Ensemble models, and various combinations of different Deep-learning models. This comprehensive strategy aims to effectively determine the sentiment conveyed in Sinhala language content. The Figure1 gives an overview of the approach used in this work.

A. Raw Data

The Sinhala Offensive Language Dataset (SOLD) [16] is a manually annotated dataset containing 10,000 posts from Twitter annotated as offensive and not offensive at both sentence level and token level. The dataset was created to address the lack of annotated data for offensive language identification in Sinhala, a low-resource Indo-Aryan language spoken by over 17 million people in Sri Lanka. Figure2 shows a snapshot of the SOLD dataset obtained from the Huggingface repository.

B. Data Preprocessing

Data preprocessing is a crucial step in preparing textual information for analysis, particularly in the context of hate speech detection. This section outlines the specific preprocessing techniques employed in our study to enhance the quality of the Sinhala language dataset.

1) Removing non Sinhalese characters: To ensure the integrity of the Sinhala language dataset, the removal of non-Sinhalese characters is imperative. This step involves filtering out characters that do not belong to the Sinhala script, thereby eliminating noise and ensuring the coherence of the textual data.

2) Removing Stopwords: Stopwords are common words that often do not contribute significant meaning to a text, and are systematically removed during this preprocessing stage. This not only reduces the dimensionality of the dataset but also focuses the analysis on content-bearing words, enhancing the efficiency of subsequent natural language processing tasks.

3) Tokenisation: involves breaking down the text into individual units, typically words or subword units. In the context of this study, tokenization is applied to segment the Sinhala text into meaningful units, facilitating subsequent feature extraction and analysis. This process serves as a foundational step in transforming the raw textual data into a format suitable for machine learning and deep learning models.

C. Word Embedding

1) Word2Vec: Using the word2vec embedding technique, a vector with a length of 100 for every word in a sentence is created. During the training of our word2vec model on the dataset, minimum word count is set to 1 and a window size to 5. To construct a sentence vector from these word vectors, all the individual word vectors are added up within the sentence and then divide the sum by the total number of words in that sentence. Essentially, this method entails computing the average of the word vectors in a given sentence. Therefore, for a sentence s the word vector v_s is defined as:

$$v_s = 1/n \sum_{i}^{n} v_i$$

where v_i is the word vector of each word present in the sentence.

2) TF-IDF: TF-IDF (Term Frequency-Inverse Document Frequency) is applied on the cleaned and preprocessed text to transform them into word embeddings. This TF-IDF transformation serves as a method to represent the significance of each word in the context of the entire dataset. The resulting TF-IDF word embeddings are then employed as input for machine learning (ML) and deep learning (DL) models.

D. Machine Learning Model

We have implemented various machine learning models such as Random Forest, Logistic Regression, Support Vector Machines, Xtreme Gradient Boosting etc. and noted the accuracy achieved in each case we have executed an array of machine learning models to assess their efficacy in predicting hate speech within the Sinhala language dataset. We implement a total of 9 Machine Learning models. The models implemented include Decision Tree, Random Forest, Logistic Regression, Support Vector Machines (SVM), Gradient Boosting, Xtreme Gradient Boosting (XGBoost), K-Nearest Neighbours, Adaptive Boosting (AdaBoost)

TABLE I
TRAIN, VALIDATION AND TEST ACCURACIES FOR MACHINE LEARNING MODEL

Model	Train Acc	Validation Acc	Test Acc	F1 Score
Logistic Regression	**0.789844**	**0.683333**	**0.684000**	**0.574506**
XGBoost	0.879687	0.680208	0.612667	0.541436
Decision Tree	0.996094	0.622917	0.669333	0.532075
Random Forest	0.996094	0.697917	0.572271	0.556604
SVM	0.945312	0.683333	0.575333	0.154050
K-Nearest Neighbors	0.613021	0.587500	0.476000	0.570022
Gaussian Naive Bayes	0.703906	0.486458	0.598667	0.528951
AdaBoost	0.717187	0.678125	0.672000	0.508000
Gradient Boosting	0.753906	0.690625	0.664000	0.550000

E. Deep Learning Model

In our study of hate speech detection in Sinhala, we employ a range of deep learning architectures designed to understand language nuances and effectively identify problematic content.

1) ANN: Artificial Neural Networks (ANNs) along with TF-IDF embedding, contribute to the model's ability to predict hate speech. hidden_layer_sizes: (100) - One hidden layer with 100 neurons, activation: 'relu' - Rectified Linear Unit activation function, solver: 'adam' - Stochastic Gradient Descent (SGD) with adaptive learning rates, alpha: 0.0001 - L2 regularization term.

2) LSTM: In our implementation, we have designed a setup that incorporates Long Short-Term Memory (LSTM) followed by varying numbers of Dense layers, along with experimentation involving different input shapes and features. Keeping the constant Architecture we tried 2 methods-
(a) We used the direct text embeddigs as input feature for the first variation of this architecture.
(b) We used Word2vec embeddings as input featues for the second variation of this architecture.
This intentional exploration of Input feature variations is conducted to systematically evaluate their influence on the accuracy and F1-Score of the model.

3) CNN + LSTM : We implemented several variations of the CNN + LSTM model. The typical architecture of this model includes an Embedding layer, followed by a Convolutional Layer, then a Dense Layer, and subsequently, an LSTM layer, followed by a variable number of additional Dense Layers.
(a) We used the direct text embeddigs as input feature for the first variation of this architecture.
(b) We used Word2vec embeddings as input featues for the second variation of this architecture.
(c)(d) We changed the Hyperparameters to check which combination gave the best results.

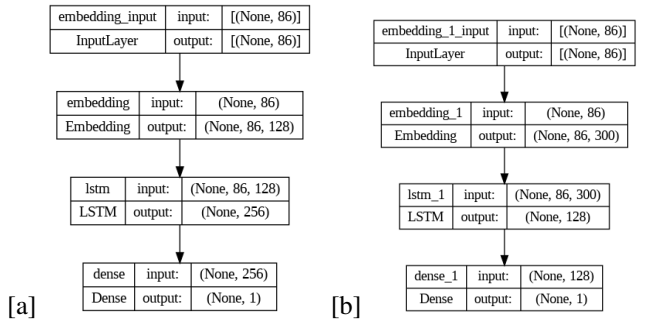

Fig. 3. LSTM Architecture with different Hyper-parameters and input features[a] LSTM [b] LSTM-Word2Vec

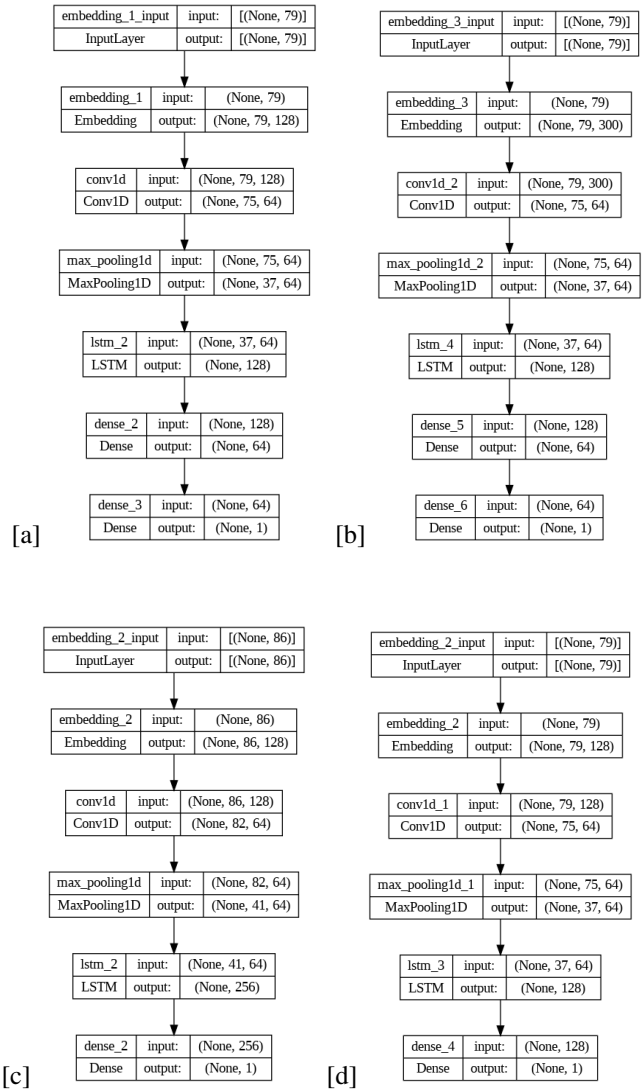

Fig. 4. CNN + LSTM Architecture [a] LSTM + CNN [b] LSTM + CNN-Word2Vec [c] LSTM + CNN 1 [d] LSTM + CNN 2

4) Stacked LSTM: We implemented 2 variations of Stacked LSTM.

(a) The typical architecture of this model includes an Embedding layer, followed by 2 LSTM layers and then a Dense Layer.

(b) The typical architecture of this model includes an Embedding layer, followed by an LSTM layer, a Dense layer, followed by anLSTM and then a Dense Layer.

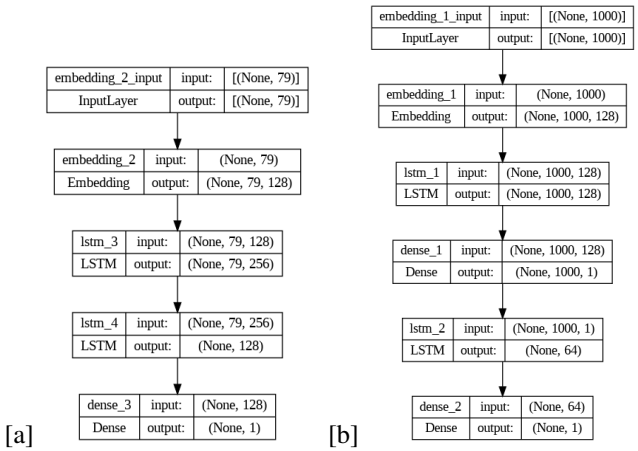

Fig. 5. Stacked LSTM Architecture [a] Stacked LSTM 1 [b] Stacked LSTM 2

5) Pretrained model RoBERTa: A RoBERTa model pretrained on the context of Sinhala language was used to get better accuracy as this increases the knowledge of the model in the context of sinhala language. RoBERTa -1, RoBERTa -2 are models trained over the existing pretrained model with differences in hyperparameters like Learning rate.

6) BERT + CNN + LSTM : The BERT + CNN + LSTM model is a proposed architecture that combines the strengths of all three models. We have used the pre-trained model RoBERTa and implemented a Convolutional Layer on it followed by an LSTM and a Dense Layer.

7) Performance Evaluation: We measure the performance of our model based on accuracy and F1 score.

IV. Results and Discussion

In this section, we present a comprehensive overview of the outcomes derived from our diverse array of Machine Learning (ML) and Deep Learning (DL) models. The focal point of our investigation is the classification of sentiment utilizing word2vec embeddings, with a keen exploration into various ML techniques. The tabulated results in Table I delineate the accuracies attained by different models, revealing intriguing patterns. It is noteworthy that Logistic Regression emerges as the top performer, achieving the highest accuracy and F1 score at 68.4% and 57.4%, respectively. Despite tendencies of overfitting in the training data, models such as Decision Tree and AdaBoost exhibit test accuracies comparable to the optimal Logistic Regression model.

Shifting our focus to DL models, our exploration reveals interesting nuances in accuracy and F1 scores. A simplistic Long Short-Term Memory (LSTM) model, illustrated in FigIII-E2, attains a commendable 73.4% accuracy. However, when integrating this LSTM with Convolutional Neural Network (CNN) models, as depicted in FigIII-E3, a drop in

TABLE II
ACCURACY AND F1 SCORE FOR DEEP LEARNING MODEL

Model	Test Acc	F1 Score
Neural Network	0.6343	0.5796
LSTM	0.7347	0.6937
LSTM-Word2Vec	0.5773	0.4097
LSTM + CNN	0.7293	0.6577
LSTM + CNN-Word2Vec	0.5760	0.1783
LSTM + CNN 1	0.7227	0.6858
LSTM + CNN 2	0.7280	0.6667
Stacked LSTM 1	0.7447	0.7190
Stacked LSTM 2	0.5620	0.0000
RoBERTa -1	0.7940	0.7810
RoBERTa -2	0.6193	0.3787
RoBERTa + CNN + LSTM	**0.8007**	**0.7555**

accuracy and F1 score is observed. Further experimentation involving stacked LSTMs, as illustrated in FigIII-E4, results in a significant accuracy decline, with F1 scores reaching zero for certain configurations FigIII-E4[b]. Notably, the most promising results were achieved through the utilization of the BERT model. However, it is imperative to acknowledge the sensitivity of the BERT model to hyperparameters, indicating the need for careful tuning in its application. Further, we find that combining the BERT model with LSTM and CNN leads to a higher accuracy and F1 score. Using the RoBERTa + CNN + LSTM model we get the best accuracy of 80.07% and F1 score of 75.55%. This is the highest accuracy achieved by all models compared in this study. In summary, our investigation elucidates the performance nuances across a spectrum of ML and DL models, providing insights into their strengths, weaknesses, and the impact of architectural choices on sentiment classification accuracy.

This provides answers to our research questions:

Answer to RQ1: Our approach to detecting sentiment in Sinhala text proves to be highly effective, utilizing word2vec for word embedding in conjunction with a combination of LSTM-CNN-BERT and a prediction model. The synergy of these elements resulted in an impressive accuracy exceeding 80%.

Answer to RQ2: The architecture of a composite deep learning model significantly influences its accuracy. Notably, we observed instances where the F1 score for certain models declined to 0. However, through meticulous adjustments of hyperparameters and modifications to the model structure, we were able to achieve substantially higher accuracy than what each individual model could attain. This highlights the importance of thoughtful model design and parameter tuning in maximizing the overall effectiveness of combined deep learning models for tasks such as sentiment analysis.

However, we would also shed light on the fact that simply using a more complex and deeper model may not solve the challenge of dealing with sentiment analysis in low resource languages. We have to find methods to deal with the scarcity of quality data. Semi-supervised machine learning techniques can be a way to advance sentiment analysis in low-resource languages, such as Sinhala. The inherent scarcity of labelled data in these languages often hampers the development of robust sentiment analysis models. Semi-supervised learning, by leveraging both labelled and unlabeled data, offers an innovative solution to this challenge. Unlabeled data, abundant but typically underutilized, can be employed to augment the training process, enhancing the model's understanding of the linguistic nuances and sentiment patterns specific to the low-resource language. Sources such as social media, textbooks religious texts etc can be used as a source for unlabeled data. In the context of Sinhala sentiment analysis, semi-supervised learning becomes particularly valuable. By incorporating a combination of labelled and unlabeled Sinhala text, the model gains exposure to a broader spectrum of language usage, thus improving its generalization capabilities. Additionally, semi-supervised techniques allow for the adaptation of pre-trained models, which may have been trained in more resourced languages, to the specific characteristics of Sinhala sentiment expression. Using Semi-supervised learning techniques such as the student-teacher model can be a way to move forward with our approach. In future, We can utilise our combined BERT+CNN+LSTM model as a "teacher" model to generate pseudo labels for the unlabeled data and train more efficient models using the true labels and the pseudo labels. These pseudo-labels can allow us to train better semi-supervised models

V. Conclusion

In this investigation, we delved into the challenge of discerning sentiments in the low-resource language of Sinhala. Through rigorous testing of various machine learning and deep learning models, we aimed to identify the most effective approaches. The Sinhala Offensive Language Dataset (SOLD) served as the basis for training and evaluating our models. Our findings revealed that, among the machine learning models, Logistic Regression outperformed others, achieving the highest accuracy at 68.40%. In the realm of deep learning models, the combination of BERT, CNN, and LSTM stood out with the highest accuracy of 80.07%. It is noteworthy that the arrangement of different models and their corresponding hyperparameters significantly influenced the accuracy outcomes. These insights emphasize the importance of thoughtful model selection and parameter tuning in achieving optimal results for sentiment classification in low-resource languages like Sinhala.

Acknowledgment

I extend my sincere thanks to Dr. Animesh Chaturvedi for his guidance and unwavering support throughout this project.

References

[1] Abdulaziz M Alayba, Vasile Palade, Matthew England, and Rahat Iqbal. A combined cnn and lstm model for arabic sentiment analysis. In *Machine Learning and Knowledge Extraction: Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Hamburg, Germany, August 27–30, 2018, Proceedings 2*, pages 179–191. Springer, 2018.

[2] Ranjan Kumar Behera, Monalisa Jena, Santanu Kumar Rath, and Sanjay Misra. Co-lstm: Convolutional lstm model for sentiment analysis in social big data. *Information Processing & Management*, 58(1):102435, 2021.

[3] PDT Chathuranga, SAS Lorensuhewa, and MAL Kalyani. Sinhala sentiment analysis using corpus based sentiment lexicon. In *2019 19th international conference on advances in ICT for emerging regions (ICTer)*, volume 250, pages 1–7. IEEE, 2019.

[4] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[5] Nisansa de Silva. Sinhala text classification: observations from the perspective of a resource poor language. *ResearchGate*, 2015.

[6] Piyumal Demotte, Lahiru Senevirathne, Binod Karunanayake, Udyogi Munasinghe, and Surangika Ranathunga. Sentiment analysis of sinhala news comments using sentence-state lstm networks. In *2020 Moratuwa Engineering Research Conference (MERCon)*, pages 283–288. IEEE, 2020.

[7] Vinura Dhananjaya, Piyumal Demotte, Surangika Ranathunga, and Sanath Jayasena. Bertifying sinhala–a comprehensive analysis of pre-trained language models for sinhala text classification. *arXiv preprint arXiv:2208.07864*, 2022.

[8] WSS Fernando, Ruvan Weerasinghe, and ERAD Bandara. Sinhala hate speech detection in social media using machine learning and deep learning. In *2022 22nd International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 166–171. IEEE, 2022.

[9] SVS Gunasekara and Prasanna S Haddela. Context aware stopwords for sinhala text classification. In *2018 National Information Technology Conference (NITC)*, pages 1–6. IEEE, 2018.

[10] Raviraj Joshi. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*, 2022.

[11] Raviraj Joshi. L3cube-mahacorpus and mahabert: Marathi monolingual corpus, marathi bert language models, and resources. *arXiv preprint arXiv:2202.01159*, 2022.

[12] Binod Karunanayake, Udyogi Munasinghe, Piyumal Demotte, Lahiru Senevirathne, and Surangika Ranathunga. Sinhala sentiment lexicon generation using word similarity. In *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 77–82. IEEE, 2020.

[13] Dimuthu Lakmal, Surangika Ranathunga, Saman Peramuna, and Indu Herath. Word embedding evaluation for sinhala. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1874–1881, 2020.

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[15] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pages 4513–4519, 2021.

[16] Tharindu Ranasinghe, Isuri Anuradha, Damith Premasiri, Kanishka Silva, Hansi Hettiarachchi, Lasitha Uyangodage, and Marcos Zampieri. Sold: Sinhala offensive language dataset. *arXiv preprint arXiv:2212.00851*, 2022.

[17] Supun Tharaka Sandaruwan, Susil Aruna Shantha Lorensuhewa, and Kalyani Munasinghe. Identification of abusive sinhala comments in social media using text mining and machine learning techniques. *The International Journal on Advances in ICT for Emerging Regions*, 13(1), 2020.

[18] Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. Dimensional sentiment analysis using a regional cnn-lstm model. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 225–230, 2016.

[19] Gihan Weeraprameshwara, Vihanga Jayawickrama, Nisansa de Silva, and Yudhanjaya Wijeratne. Sentiment analysis with deep learning models: a comparative study on a decade of sinhala language facebook data. In *Proceedings of the 2022 3rd International Conference on Artificial Intelligence in Electronics Engineering*, pages 16–22, 2022.