ISSN: 2182-2069 / E-ISSN: 2182-2077

Integrating Hybrid Neural Networks and Domain-Specific Embeddings for Detecting Hate Content in Code Mixed Social Media Comments

L.K. Dhanya[1*], and Dr. Kannan Balakrishnan[2]

[1*]Department of Computer Applications, Cochin University of Science and Technology, Kochi, Kerala, India;
Mar Baselios College of Engineering and Technology, Trivandrum, Kerala, India.
dhanya.lk@cusat.ac.in, https://orcid.org/0000-0002-1383-4384

[2]Department of Computer Applications, Cochin University of Science and Technology, Kochi, Kerala, India. bkannan@cusat.ac.in, https://orcid.org/0000-0003-1285-5949

Received: April 04, 2024; Revised: June 10, 2024; Accepted: July 05, 2024; Published: August 30, 2024

Abstract

Hate speech is any communication intended to irritate, intimidate, disrupt, or incite anger in an individual or a group, typically targeting characteristics such as religion, ethnicity, appearance, or sexual orientation. Inhabitants of multilingual communities often engage in conversations using multiple regional languages. This sort of textual communication is known as code-mixed data since it combines many languages. This research shows how to recognize and detect hate speech in code-mixed Malayalam-English (Manglish) material. We created a dataset of Manglish-written social media comments from platforms like YouTube and Facebook. Before delving into word embeddings, we developed a unique stopword list designed specifically for Manglish, which has never been done previously. This bespoke stopword list significantly enhanced our data preparation operations. Following that, we concentrated on evaluating several word embedding techniques. We then utilized Glove to develop a distinct domain-specific word embedding model (DSG)for Malayalam-English code-mixed data. This concept was crucial in increasing the overall efficiency of our model. In addition to the approaches described above, we conducted a comprehensive set of experiments using several classifiers such as logistic regression, SVM, and XGBoost, as well as deep learning models such as Convolutional Neural Network (CNN) and bidirectional Long-Short-Term Memory (BiLSTM). Following thorough experimental testing, we suggested a unique hybrid deep-learning model with domain-specific word embeddings. This technique was quite effective in managing our dataset, with an astonishing 96.4% accuracy in detecting hate speech in Manglish comments.

Keywords: Hate Speech, Domain Specific Word Embedding, Hybrid Deep Learning, Stopwords, Code-mixed Data, Manglish.

Journal of Internet Services and Information Security (JISIS), volume: 14, number: 3 (August), pp. 316-329.
DOI: *10.58346/JISIS.2024.I3.019*

*Corresponding author: Department of Computer Applications, Cochin University of Science and Technology, Kochi, Kerala, India; Mar Baselios College of Engineering and Technology, Trivandrum, Kerala, India.

1 Introduction

The rise of social networking and communication technology has led to increased spread of hate content, which can lead to mental health issues like depression, insomnia, and suicidal ideation over time (Paz et al., 2020). Detecting hate speech is challenging due to user-generated multilingual content and code-mixing issues when one language is entered in transliterated format (Anbukkarasi & Varadhaganapathy, 2023) Malayalam, spoken by about 2.88% of India's population and designated as a "Classical Language of India" in 2013 (Sebastian, 2023), is frequently blended with English to create Manglish. Manglish involves code-mixing, in which speakers seamlessly move between Malayalam and English, resulting in a distinct linguistic style. Code-mixing may be divided into two. Intra-sentential code-mixing happens when speakers use components of two or more languages in the same phrase. In contrast, inter-sentential code-mixing includes switching languages between sentences or utterances. Intra-word code-mixing is distinguished by the incorporation of components from many languages into a single word. Table 1 shows instances of various code-mixing kinds and their meanings.

Table 1: Examples of Manglish Sentences and their Types

Manglish Sentence	English Meaning	Type
Oh my god! enth sambhavichu?	Oh my god! What happened?	Inter sentential code mixing
nee happy aano?	Are you happy?	Intra sentential code mixing
trophikal	trophies	intra word code mixing

This study aims to solve the issues of identifying hate speech in Manglish, while also portraying the region's cultural and linguistic diversity. As a result, the current study explores whether or not texts in a corpus are offensive by employing a variety of traditional machine learning classifiers and deep learning models (Risch & Krestel, 2019). The integration of various deep learning models and the construction of domain-specific word embedding are the innovative aspects of this research work. The current study makes use of a recently created dataset derived from social media networks (Buchag et al., 2022). Section 3.1 will through the specifics of the dataset (Vimitha & Gireesh, 2024). This dataset will be one of the study work's unique features. We address the following research questions with this study:

RQ1: Does pre-processing affect machine learning algorithms for code-mixed Malayalam-English data?

RQ2: Do the currently available word embedding models perform effectively with Malayalam-English code-mixed data?

RQ3: Which model does well in detecting hate speech in Malayalam-English code-mixed data?

The main contributions of this study are as follows:

1. The research involves the creation of a new annotated dataset containing 10,000 comments, which serves as valuable data for the investigation.
2. As part of this work a new stop word list specific to Malayalam-English (Manglish) code-mixed text was createdwhich enhanced the quality of the analysis.
3. The study represents the new attempt to detect cyberbullying in Manglish data using a hybrid deep learning model, highlighting its innovative approach. The use of a Hybrid deep learning model along with domain-specific word embedding in Malayalam-English code-mixed data to identify cyberbullying text, marking the first-ever effort in this area.

4. The research introduces the generation of a domain-specific word embedding, named "glove. model_ MANGLISH.model," from Malayalam-English code-mixed data, a novel contribution in the field.

The structure of this paper is outlined as follows: In Section 2, we provide a brief overview of the existing literature. In Section 3, we present our proposed methodology. Extensive experiments and a thorough analysis of the results are presented in Section 4 followed by concluding remarks in Section 5.

2 Related Works

Our research methodology focused on curating papers from 2018 to 2023 related to hate speech detection in Indian languages and code-mixed data, aiming to develop algorithms for Malayalam-English code-mixed content. Social media platforms are overwhelmed with hate speech, necessitating automated filtering due to the impracticality of manual sorting (Surendar et al., 2024). While many studies have focused on English, some have utilized lexical approaches (Wang et al., 2022) or traditional machine learning (Abro et al., 2020), which lack contextual understanding and are prone to adversarial attacks. Consequently, deep learning models, including RNNs and hybrid models, have gained traction for their ability to autonomously extract relevant features (Akila & Revathi, 2023). Minimal research has been done on Indian languages (Dhanya & Balakrishnan, 2021), with studies mostly concentrating on Hindi and Bengali, employing models like SVM and Random Forest. Recent efforts also include the exploration of Marathi using deep learning architectures such as CNN, LSTM, and transformer-based models (Choi & Zhang, 2022). Our survey on Indian code-mixed languages revealed a focus on Hinglish (Yadav et al., 2023), with studies using approaches like transfer learning, data augmentation, and a variety of classifiers. Some notable methods include FE-DGRNN for multilingual texts (Ayo et al., 2021) and BiGRU combined with TF-IDF for aggressive tweet detection

3 Proposed Methodology

Initially the comments are in a text format. During a preprocessing stage, unnecessary special characters are eliminated. After the data has been purified, it undergoes further preprocessing processes, including the removal of stop words and other techniques. In order to identify important characteristics, we employ traditional word embedding methods like as Word2vec and Glove, together with the well-established TF-IDF methodology. In addition, LSTM models that operate at the sub-word level and word embeddings that are specialized to a certain domain have shown effectiveness in identifying hate speech in Hinglish. This comprehensive assessment emphasizes the significance of specialized techniques for identifying hate speech in multilingual and code-mixed comments..

3.1 Data set

Data collection and annotation provide substantial obstacles to training artificial classifiers to detect hate speech.Because there is no commonly accepted definition of hate speech, it is difficult to precisely identify and label specific texts (Wang et al., 2022). As a result, there are few datasets available for public use, with Twitter serving as the primary source due to its more permissive data usage policy. However, the usefulness of Twitter resources is limited by the platform’s distinct nature, which is defined by character limits and terse, short-form material. Other platforms, on the other hand, provide longer posts, providing a distinct dataset for study. Architecture of Proposed Method shown in Figure 1.

In this study, we used YouTube videos to generate a code-mixed corpus of Malayalam-English. We have collected videos from renowned Malayalam news channels like as Mathrubhumi, 24 News, and Manorama covering topics such as politics religion and cinema. Because of their popularity, these videos received a great amount of comments and reactions. Using the YouTube Data API, we gathered detailed information on each video, including its ID, title, comments, replies to comments, likes, date, and time. We stored the extracted comments and replies from each video in separate CSV files in chronological order. Eventually, we consolidated the dataset into a single CSV file containing only the “comments” column. However, we removed the portion where 70% of the comments were in a single language (Malayalam or English) to focus on the remaining comments, which are either multilingual (Manglish) or a mix of Malayalam and English. Sample comments with labels from the dataset are given in Table 2.

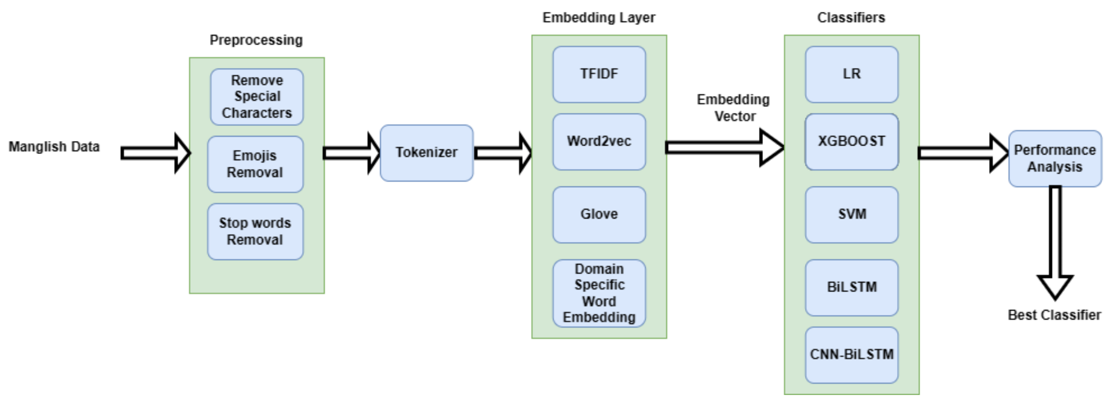

Figure 1: Architecture of Proposed Method

Data annotation, also known as data categorization and labelling for AI applications, is critical in training data for various use cases. It entails accurately identifying and annotating training data, which frequently necessitates coordination between annotators with language comprehension and context comprehension. Cohen’s kappa coefficient (Więckowska et al., 2022) is an important measure for gauging annotator dependability since it takes into account the likelihood of agreement occurring by chance and gives a more robust assessment than simple percentage agreement computations for qualitativeitems.

Table 2: Sample Social Media Comments from the Dataset

Sentence	**Meaning**	**Label**
Veenayude questions nerikettathum nilavaram illathavayumanu	Veena’s questions are crude and substandard	HATE
Idak keri paranjapo pulli sorry paranje ketto	Did you hear him say sorry When speech was interrupted	NHATE
Nanam kettavan and swontham mole nokkathathavan	He is the one who doesn’t look after his own daughter and shameless person	HATE

In this study, two annotators with linguistic expertise in both Malayalam and English manually annotated the datasetidentify hate speech. Cohen’s Kappa was used to assess inter-annotator agreement (Więckowska et al., 2022) for hate speech annotations across two sets of 10000 code-mixed texts, yielding a Kappa score of 0.908, indicating high-quality annotations. In the study, hate speech is labelled as "HATE" while non-hateful messages are labelled as "NHATE". In this dataset,hate speech comments constitute only 37% of the total, while non-hate speech comments account for the remaining63%. This creates an imbalance during the training phase, as illustrated in Figure 2.

If the amount of hate values in our dataset is similar to the quantity of non-hate values, we can consider our dataset balanced; otherwise, it is called unbalanced. We have 6300 non-hate categories and 3700 hate categories in our scenario. To correct the imbalance, we used the Synthetic Minority Oversampling Technique (SMOTE) (Fernández et al., 2018) which creates new synthetic cases intelligently from the minority class (hatred) subset of the data and incorporates them into the existing dataset. This technique mitigates the impact of a small number of minority class incidents. SMOTE balances the class distribution by generating synthetic samples for the minority class (the class with fewer instances). This is accomplished by generating new samples that are comparable to the existing minority class instances. Selecting a minority class instance, locating its k nearest neighbours (usually using Euclidean distance), and then constructing synthetic instances by interpolating between the chosen instance and its neighbours are all steps in the process.

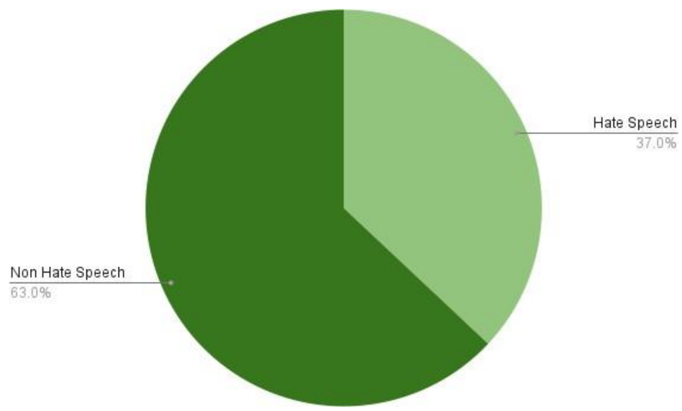

Figure 2: Pie Chart Illustrating the Imbalance of the Original Training Dataset

3.2 Data Pre-processing

This section provides a concise overview of the preprocessing techniques applied to the Manglish code mixed data. Code-mixed data is collected from multiple sources and often contains a considerable amount of noise, includingpunctuation, white spaces, numbers, special characters, extra spaces, emojis, and stop words (Tabassum & Patil, 2020). This raw code-mixed data presents challenges for the analysis process. Therefore, it is necessary to undergo data preprocessing to transformthe raw data into a cleaner format. Data preprocessing is essential to ensure that the data is properly formatted, enabling more effective results when utilizing this processed data in various models. In the realm of hate speech identification research, there has been relatively less emphasis on data "cleaning" compared to other natural language tasks (Tabassum & Patil, 2020). This is likely due to the complex nature of hate speech language, which demands a deeper level of analysis than standard text. Some users cleverly circumvent platform speech restrictions by substituting letters with symbols to covertly convey otherwise prohibited messages. Researchers often employ common strategies like converting textto lowercase, tokenization of tweets, and removing URLs. Emojis, although potentially valuable for enhancing NLP task performance, are frequently excluded. In the system, we implemented the following preprocessing methods for code-mixed data. here we can find a comprehensive illustration of these techniques in Figure 3.

We have created a specific stop word list for Manglish (Malayalam-English), called SW_MANGLISH, which is now available for further study. The NLTK package was utilized to include our unique stop word list into its corpus, resulting in better data preparation. Table 3 shows the sample stop word list.

3.3 Proposed Domain-Specific Word Embedding

Feature extraction is critical in NLP and machine learning, with approaches such as TF-IDF, Word2Vec, and GloVe being particularly useful. Word2Vec (Ma & Zhang, 2015) depicts words as dense vectors based on co-occurrence, whereas GloVe (Pennington et al., 2024) collects syntactic and semantic similarities utilizing global and local co-occurrence data, resulting in a com- prehensive word meaning representation. In this work, GloVe was utilized to generate domain-specific word embeddings to supplement pre-trained GloVe embeddings and improve text analysis and model performance in NLP applications. We discovered difficulties when utilizing generic pre-trained word embedding algorithms to detect hate speech in code-mixed Malayalam-English text. The existing pre-trained embeddings may not function optimally due to the difficulties of code-mixing and the specific peculiarities of our dataset.

Table 3: Sample Stop Word List

Stop Word	Meaning in English
valare	Very much
valiya	Big
cheriya	Small
ivite	Over here
avite	Over there
engane	How
evide	Where
eppol	When

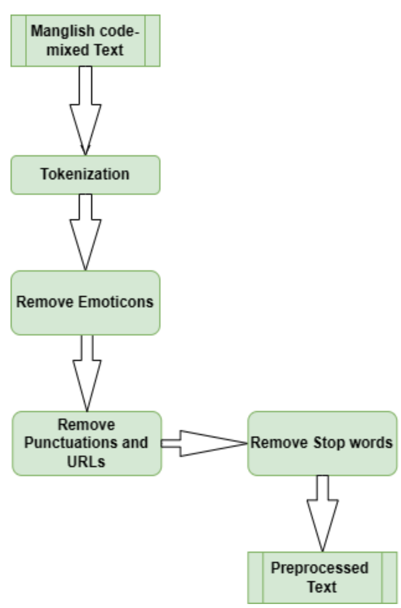

Figure 3: Pre-Processing of Manglish Code-mixed Text

In order to address this issue, we proactively developed novel domain-specific word embeddings specifically tailored for Manglish (a combination of Malayalam and English) data. To effectively address the intricacies of code-mixing and context in our dataset, we may enhance our hate speech detection model by creating domain-specific word embeddings. By adapting the word embeddings to the specific attributes of our dataset, we are showcasing a deliberate and data-oriented strategy for tackling the distinct difficulties presented by code-mixed text in languages with limited resources. Sample Word Cloud shown in figure 4, and Figure 5 provides a detailed explanation of the procedure for developing a novel embedding model called Domain Specific Glove (DSG), which is then saved as glove.model_MANGLISH.model.

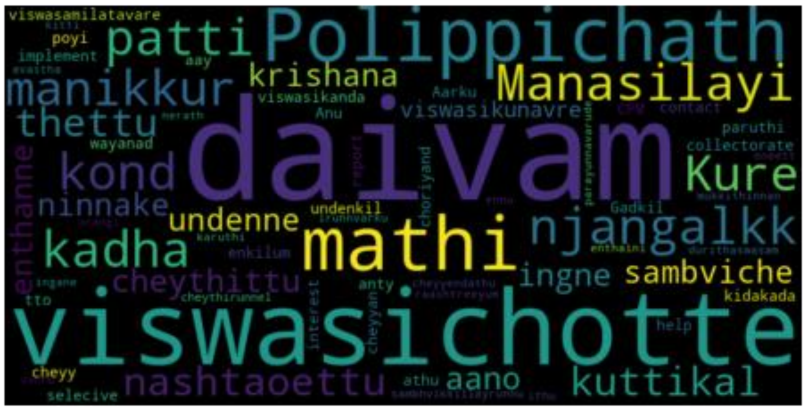

Figure 4: Sample Word Cloud

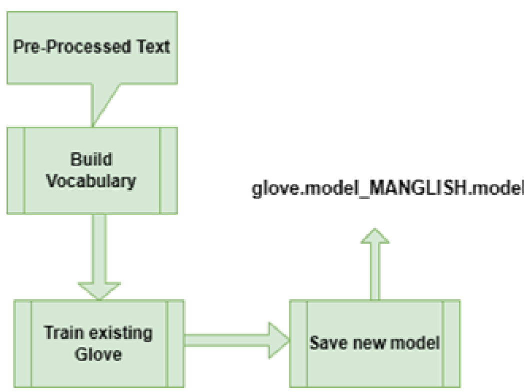

Figure 5: Methodology to Create DSG

3.4 Proposed Classification Model

In this study, models are utilized to train on embedded data, functioning as binary classifiers to categorize the embedded data into ’NHATE’ or ’HATE’. Since the datasets are inherently noisydue to their social media origin, preprocessing is crucial and was conducted at the beginning of this research work. This study looked at a number of hate speech categorization models, including Logistic Regression, SVM, XGBoost, and BiLSTM, and focused on their skills in binary classification, margin optimization, parallel tree boosting, and contextual information gathering. The study underlines the necessity for classical, deep learning, and hybrid models trained on preprocessed data to successfully detect hate speech on social media. Recurrent Neural Networks (RNNs) (Nowak et al., 2017) are often used for sequential data, but Long Short-Term Memory (LSTM)

(Nowak et al., 2017) networks are a variant that includes input, forget, and output gates, as well as memory cells for data storage.

The CNN-BiLSTM model was trained using the binary cross-entropy loss function and the Adam optimizer. These options are commonly used for binary classification problems and perform well in deep learning models. Testing the model over multiple epoch values is critical for attaining the best trade-off between training accuracy and generalization performance (validation accuracy). Overfitting is a typical issue in deep learning, in which the model performs exceptionally well on training data but fails to generalize to new data. Based on the data, it was determined that the best number of epochs is between 5 and 50. This range shows that the model performs satisfactorily within this iteration range. Specifically, it leads to a validation and training accuracy gap of 0.13 to 0.17 points. This gap signifies the difference between how well the model performs on the training data and how well it generalizes to new, unseen data.

4 Results and Discussion

In this section, we describe the experiments carried out to evaluate the performance of our proposed models, which include traditional machine learning methods (XGBOOST, SVM, LR), a deep learning model (BiLSTM), and a hybrid model (CNN+BiLSTM). We present the results of 20 sets of experiments conducted using the preprocessed Malayalam-English code-mixed dataset discussed in Section 4.1, where various text-processing techniques were applied. Machine learning and deep learning models were applied in Google Colab using Python libraries such as Keras with the TensorFlow backend, NumPy, NLTK, and Scikit-learn. To assess how well the model performs, metrics like Precision (Eberhart et al., 1990), Recall (Eberhart et al., 1990), and F1-score (Eberhart et al., 1990) were utilized. Precision refers to the model's ability to correctly identify offensive messages among those it predicted as offensive. Recall, on the other hand, measures the model's capability to correctly identify offensive messages out of all the actual offensive messages. The F1-score, which is the harmonic mean of precision and recall, provides a comprehensive evaluation of the model's performance in handling offensive contentdetection shown in figure 6.

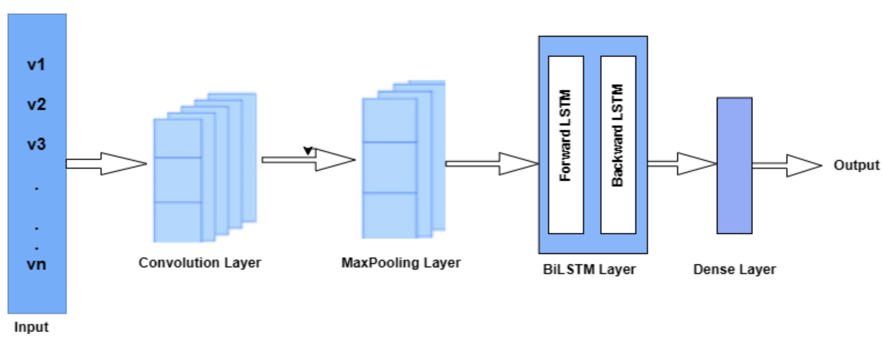

Figure 6: Proposed Classifier

Table 4 contains the classification report for the Malayalam-English code-mixed dataset, encompassing a range of traditional machine learning, deep learning and hybrid deep learning models. Figure 7 shows classification report of all models. Among all the machine learning modelsincluded in this study, the XGBOOST model obtained the highest accuracy of 85.19% with the DSG (domain-specific word Embedding using Glove). When compared with the deep learning model and hybrid model, Our proposed hybridmodel CNN-BiLSTM with DSG got the highest accuracy of 96.4 %.

In a balanced dataset (where the number of instances in each class is roughly equal), accuracy and the F1 score can be expected to increase in a similar manner as the model's performance improves. When the model correctly predicts both positive and negative instances, both metrics increase. The following are the objectives of this study.

- To identify the optimal feature embedding technique for code mixed comments.
- To ascertain the most effective classification technique.

Table 4: Performance Analysis of Various Models

Model	Precision	Recall	F1-score	Accuracy
LR+TFIDF	80.68	71.69	69.45	71.69
SVM+TFIDF	79.99	67.16	63.23	67.16
XGBOOST+TFIDF	84.21	80.16	79.55	80.16
BiLSTM+TFIDF	78.41	63.17	57.46	63.17
CNN-BiLSTM+TFIDF	81.57	70.91	68.23	70.91
LR+Word2Vec	81.18	70.64	67.45	70.64
SVM+Word2Vec	81.99	69.16	61.23	69.16
XGBOOST+Word2Vec	84.51	82.16	79.25	82.16
BiLSTM+Word2Vec	86.41	83.17	65.46	83.17
CNN-BiLSTM+Word2Vec	88.57	87.91	75.23	87.91
LR+Glove	83.01	72.64	69.45	72.66
SVM+Glove	82.01	71.16	64.23	71.16
XGBOOST+Glove	85.00	84.17	80.25	84.17
BiLSTM+Glove	88.41	86.17	73.36	86.17
CNN-BiLSTM+Glove	90.23	89.45	77.34	89.45
LR+DSG	84.33	77.64	72.78	77.64
SVM+DSG	84.01	75.16	68.23	75.16
XGBOOST+DSG	87.00	85.19	81.45	85.19
BiLSTM+DSG	92.45	89.93	80.01	89.93
CNN-BiLSTM+DSG	97.68	96.4	84.02	**96.4**

4.1 Finding the Suitable Embedding Technique

The selection of the suitable embedding strategy in natural language processing (NLP) is crucial as it significantly impacts the performance, interpretability, and ability to generalize to other types of data. We have employed four distinct embedding methodologies: TFIDF (Kumar & Subba, 2020), Word2vec (Biswas & De, 2022), Glove (Biswas & De, 2022), and DSG (Domain Specific Word embedding using Glove). Figure 8 compares different word embedding techniques.

Figure 7: Confusion Matrix for all the Models

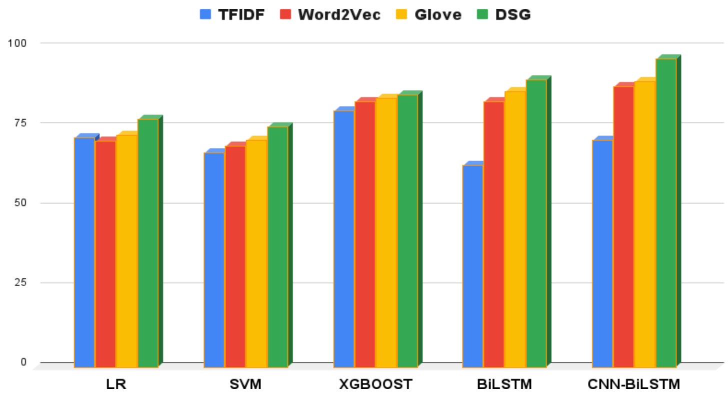

Figure 8: Comparison of Embedding Techniques

4.2 Finding the Suitable Classification Model

We conducted a series of experiments to identify the most effective approach for predicting hate speech in a code-mixed dataset. Initially, we explored traditional machine learning (ML) techniques, including LR (Logistic Regression), XGB (XGBoost), and SVM (Support Vector Machine). Subsequently, we investigated deep learning (DL) models, specifically BiLSTM (Bidirectional Long Short-Term Memory), and introduced a novel hybrid model, CNN- BiLSTM. To assess the model's performance, we employed metrics such as precision, recall, and accuracy for each class individually, as well as for the entire dataset. The model achieving the highest weighted accuracy was identified as the most suitable model for our task. The model's weights are adjusted throughout each epoch based on the measured loss and the optimization technique (Adam optimizer) to minimize the loss and enhance accuracy. We can witness a decrease in both training and validation loss and a rise in both training and validation accuracy as the epochs advance. This signifies that the model is increasing its performance by learning from the training data. The validation accuracy is continuously high, indicating good generalization. By the end of the 25 epochs, the CNN-BiLSTM model has achieved a high training accuracy of 0.960 and a high validation accuracy of 0.964. Accuracy of Various Classification Models shown in figure 9.

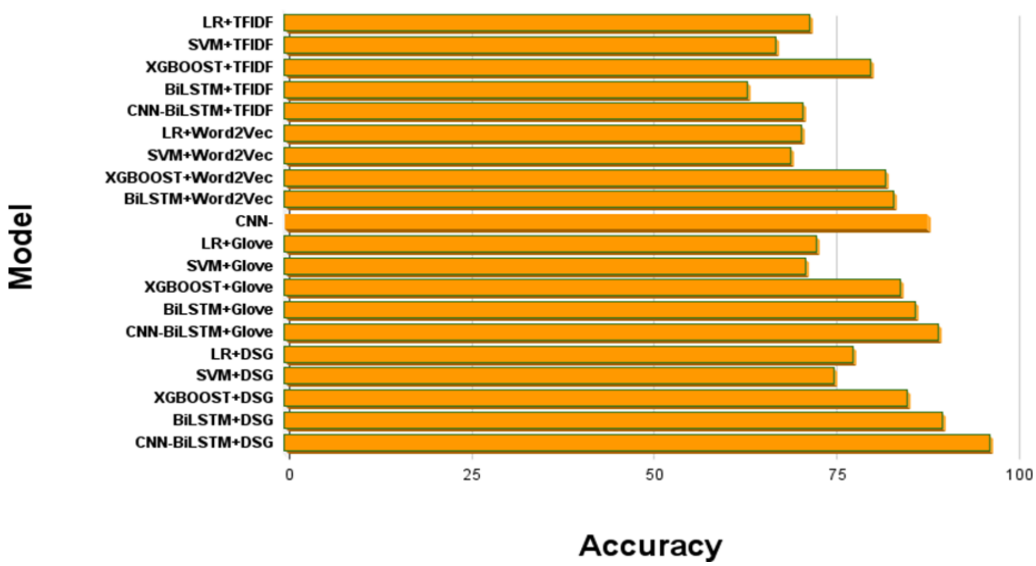

Figure 9: Accuracy of Various Classification Models

5 Conclusion and Future Work

This research utilized several deep learning models and introduced a hybrid deep learning model. This hybrid model attained an impressive accuracy rate of 96.4% in detecting hate speech. A key outcome of the study was the development of domain-specific word embeddings for the Malayalam-English code-mixed data using GloVe (Global Vectors for Word Representation). Because we are aware that the available resources are restricted, we have only used simple models for the time being. However, we will continue to develop in this area in the future. We intend to use native language processing technology to overcome linguistic obstacles. Moreover, future research projects of the Manglish hate speech detection system aim to expand their scope to include offensive speech recognition on public Facebook pages and other social media platforms. Furthermore, the future studies of the Manglish hate speech detection system aim to widen their purview to include the spotting of inflammatory speech on public Facebook pages and other social media platforms. Moreover, our objective is to create Manglish datasets that encompass spam, hate speech, and fake information. In the long run, we intend to merge machine learning and deep learning models to create a powerful ensemble model. This approach will undoubtedly aid us in enhancing our current outcomes and achieving more significant advancements in our future work.

References

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

[2] Akila, R., & Revathi, S. (2023). Fine Grained Analysis of Intention for Social Media Reviews Using Distance Measure and Deep Learning Technique. *Journal of Internet Services and Information Security*, *13*(2), 48-64.

[3] Anbukkarasi, S., & Varadhaganapathy, S. (2023). Deep learning-based hate speech detection in code-mixed Tamil text. *IETE Journal of Research*, *69*(11), 7893-7898.

[4] Ayo, F. E., Folorunso, O., Ibharalu, F. T., Osinuga, I. A., & Abayomi-Alli, A. (2021). A probabilistic clustering model for hate speech classification in twitter. *Expert Systems with Applications*, *173*, 114762. https://doi.org/10.1016/j.eswa.2021.114762

[5] Biswas, R., & De, S. (2022). A Comparative Study on Improving Word Embeddings Beyond Word2Vec and GloVe. *In IEEE Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 113-118.

[6] Buchag, R. B., Badu, A. K., Adetsi, P., & Adjei, S. (2022). Social Media Advertising for Small-Medium Scale Enterprises in Kumasi, Ghana. *Indian Journal of Information Sources and Services*, *12*(2), 28-36.

[7] Choi, J., & Zhang, X. (2022). Classifications of restricted web streaming contents based on convolutional neural network and long short-term memory (CNN-LSTM). *Journal of Internet Services and Information Security, 12*(3), 49-62.

[8] Dhanya, L. K., & Balakrishnan, K. (2021). Hate speech detection in asian languages: a survey. *In IEEE International Conference on Communication, Control and Information Sciences (ICCISc)*, *1*, 1-5.

[9] Eberhart, R. C., Dobbins, R. W., & Hutton, L. V. (1990). Performance metrics. *In Neural Network PC Tools*, 161-176.

[10] Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, 863-905.

[11] Kumar, V., & Subba, B. (2020). A Tfidf Vectorizer and SVM based sentiment analysis framework for text data corpus. *In IEEE National Conference on Communications (NCC)*, 1-6.

[12] Ma, L., & Zhang, Y. (2015). Using Word2Vec to process big text data. *In IEEE International Conference on Big Data (Big Data)*, 2895-2897.

[13] Nowak, J., Taspinar, A., & Scherer, R. (2017). LSTM recurrent neural networks for short text and sentiment classification. *In Artificial Intelligence and Soft Computing: 16th International Conference, ICAISC 2017*, 553-562.

[14] Paz, M.A., Montero-D´ıaz, J., & Moreno-Delgado, A. (2020). Hate speech: A systematized review. *Sage Open*, *10*(4), 2158244020973022. https://doi.org/10.1177/2158244020973022

[15] Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543.

[16] Risch, J., & Krestel, R. (2019). Domain-specific word embeddings for patent classification. *Data Technologies and Applications*, *53*(1), 108-122.

[17] Sebastian, M. P. (2023). Malayalam natural language processing: challenges in building a phrase-based statistical machine translation system. *ACM Transactions on Asian and Low-resource Language Information Processing*, *22*(4), 1-51.

[18] Surendar, A., Saravanakumar, V., Sindhu, S., & Arvinth, N. (2024). A Bibliometric Study of Publication- Citations in a Range of Journal Articles. *Indian Journal of Information Sources and Services*, *14*(2), 97–103. https://doi.org/10.51983/ijiss-2024.14.2.14

[19] Tabassum, A., & Patil, R. R. (2020). A survey on text pre-processing & feature extraction techniques in natural language processing. *International Research Journal of Engineering and Technology (IRJET)*, *7*(6), 4864-4867.

[20] Vimitha, R.V.L., & Gireesh, K.T. (2024). VADIA-Verkle Tree-based Approach for Dealing Data Integrity Attacks in Opportunistic Mobile Social Networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, *15*(1), 154-171.

[21] Wang, C. C., Day, M. Y., & Wu, C. L. (2022). Political hate speech detection and lexicon building: A study in Taiwan. *IEEE Access*, *10*, 44337-44346.

[22] Więckowska, B., Kubiak, K. B., Jóźwiak, P., Moryson, W., & Stawińska-Witoszyńska, B. (2022). Cohen's kappa coefficient as a measure to assess classification improvement following the addition of a new marker to a regression model. *International Journal of Environmental Research and Public Health*, *19*(16), 10213. https://doi.org/10.3390/ijerph191610213

[23] Yadav, A. K., Kumar, M., Kumar, A., Shivani, Kusum, & Yadav, D. (2023). Hate speech recognition in multilingual text: hinglish documents. *International Journal of Information Technology*, *15*(3), 1319-1331.

Authors Biography

L.K. Dhanya, works as an Assistant professor in the Department of Computer Science and Engineering at Marbaselios College of Engineering and Technology. She is currently doing PhD under the guidance of Dr Kannan B in the Department of Computer Applications, at Cochin University of Science and Technology. She has 8 research publications in various journals and conferences. Her areas of interest are natural language processing, machine learning, and deep learning.

Dr. Kannan Balakrishnan, works as an Emeritus Professor in the Department of Computer Applications, at Cochin University of Science and Technology. He received his Ph. D. in Futures Studies with specialisation in Graph Algorithms from the University of Kerala. His research interests include algorithms, graph theory, machine intelligence, image processing and complex networks. He has many papers published in national and international journals and conference proceedings. He has produced 12 PhDs in these areas. He has also published two patents.