National College of Ireland

Sentiment Analysis In Tamil Language Using Hybrid Deep Learning Approach

MSc Research Project
Data Analytics

Suba Sri Ramesh Babu
Student ID: X21100241

School of Computing
National College of Ireland

Supervisor: Taimur Hafeez

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Suba Sri Ramesh Babu
Student ID:	X21100241
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Taimur Hafeez
Submission Due Date:	15/08/2022
Project Title:	Sentiment Analysis In Tamil Language Using Hybrid Deep Learning Approach
Word Count:	8439
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Suba Sri Ramesh Babu
Date:	14th August 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Sentiment Analysis In Tamil Language Using Hybrid Deep Learning Approach

Suba Sri Ramesh Babu
X21100241

Abstract

Due to the rise of social media, the number of people using social media has also getting increased day by day from every corner of the world. This is the place for many people to share and discuss their opinion. However, these opinions were shared by various people in various languages. In recent studies, the new advancement in Machine learning and deep learning made the natural language processing task perform better in rich resources language such as English. However, these advancement has not been reached in Tamil language because of the less resources. Tamil language is one of the morphological rich language. There are a limited number of studies conducted in the field of sentiment analysis in Tamil language due to the complexity of the process and the limited resources. This research work aims to propose hybrid deep learning approaches that combines the capabilities of two different deep learning algorithms. These are the CNN-BiLSTM, CNN-LSTM, and CNN-BiGRU. In this study, to prepare the data various tools and libraries which supports Tamil language were used. The proposed methods will be evaluated and compared on various metrics such as accuracy, recall, and F1 to find the best performing model among them. The hybrid model will be able to classify the sentiments in the movie reviews in the Tamil language. The result shows that CNN-BiLSTM has achieved the higher accuracy of 80.2% and highest f1-score of 0.64 when compared to other two models.

Keywords- NLP, Sentiment Analysis, Deep Learning, Hybrid Deep learning, fasttext, word embedding, CNN-LSTM, CNN-BiLSTM, CNN-BiGRU.

1 Introduction

The proliferation of social media has allowed people to share their views on various topics, such as movies, television shows, and products. Texts can be used to provide users with factual information about these topics. Due to the large number of people using social media, the amount of information that can be presented in these types of texts has increased. The use of text as a medium to provide users with subjective information is beneficial for both expert and general users as it allows them to make informed decisions. Many companies are working on analyzing the reviews shared by peoples in order to understand their customer needs. However, extracting this information from the text is not as simple as it sounds. Some of the challenges we face when it comes to analysing the text include the introduction of new terms, the spelling mistakes of multiple languages, punctuations, repeated characters and so on. The natural language processing is a technique that goes through various steps in processing the text (Chowdhary; 2020).

The field of sentiment analysis has been growing in popularity in recent years. It is a part of Natural Language Processing, which focuses on the mapping and refinement of the opinions from the narrative. Sentiment analysis is useful in various fields because it categorizes the opinions into either positive or negative. Opinions are very important for various activities, such as getting a product or watching a movie. It can be challenging to find an opinion that is both positive and negative, as each individual has their own opinion on the product or the movie. Opinions are private expressions, which are not observed by others. The three levels of sentiment analysis are: sentence level, aspect level, and document level (Ramanathan et al.; 2021). One of the most commonly used levels of sentiment analysis is the sentence level, which we have used to analyze the review sentiment of movies. It determines whether the sentence is objective or subjective. If the sentence is objective, then the level of SA determines whether the opinion is positive or negative. Although sentiment analysis is commonly used in English, it is used less in some of the other regional languages like Tamil because of the poor resource. Despite the increasing number of people using social media, the number of people who share their opinions on their languages also increases.

1.1 Background and motivation

The oldest known language in the world is Tamil, which is spoken by around seventy eight million people globally. It is also approved as the official language in India, Srilanka, and Singapore. It is additionally one of the ancient Indian languages that follows the Subject Object Verb pattern. In addition, since Tamil is a distinct language from other Indian languages, it is not considered a standard corpora for sentiment analysis. According to (Visuwalingam et al.; 2021), Tamil is agglunative, and its structure is rich. It has root words and their corresponding suffixes. This makes its computer system complex. Due to the lack of resources and the agglunative nature of the language, very few studies have been carried out on the various aspects of sentiment analysis in Tamil. Despite the widespread use of this language, the exact techniques and mechanisms related to sentiment analysis are still not widely studied.

There are also various platforms that allow people to watch TV shows and movies. They can additionally share their opinions on social media platforms such as YouTube, Facebook, and Twitter. Everyone has their own unique structure when it comes to language. This makes it hard for the machine to process the text in different languages. For instance, in Tamil, the rich morphology of the language makes it difficult to process the text. In addition, due to the low number of resources available in Tamil, it has been hard to work on sentiment analysis in this discipline. The Tamil cinema industry, also known as Kollywood, has been drawing attention due to the fact that it started producing world-class films in recent times. This has led to the increasing number of non-Tamil speakers watching Tamil films. The Internet plays a vital role in gathering the critical reviews of the movies. The goal of this project is to analyze the movie reviews in Tamil using a hybrid deep learning approach. We will then be able to identify the positive and negative effects of the movie. Previous studies have used deep learning techniques and machine learning to classify the text. However, very few studies have been conducted on the sentiment analysis in Tamil using hybrid deep learning algorithms. In this proposed work, hybrid deep learning approach to classify the movie reviews in Tamil. Unlike, the machine learning techniques the deep learning models use the word embedding to learn the structure and meaning of the language. The hybrid models such as CNN-LSTM,

CNN-BiLSTM and CNN-BiGRU were developed and the compared to get the better accuracy giving model.

1.2 Research Question

The goal of this study is to evaluate and compare the performance of hybrid deep learning models when it comes to classifying the movie reviews which are in Tamil language.

RQ: "How well the hybrid deep learning models namely CNN-LSTM, CNN-BiLSTM and CNN-BiGRU can improve the performance of model on predicting the sentiment on Tamil Movie Review?"

1.3 Research Objectives and contribution

1.3.1 Research Objectives

This section shows the objective and contribution of the research work.

Objective 1: Explore and investigate the application of deep learning techniques used for sentiment analysis by reviewing various literatures.
Objective 2: Analyse the tools and libraries that supports Tamil language for pre-processing and feature extraction.
Objective 3: Developing Hybrid deep learning model and evaluate them.
Objective 4: Compare the outcomes of the proposed models.

1.3.2 Contribution

The development in NLP for Tamil Language has not yet reach the advancement like English or any other rich resource Languages. However, some recent studies shows interest analysing the Tamil text. This study experiment the new state-of-art hybrid deep learning techniques which can help in analysing the Tamil text and classify the sentiment. This work also gives some insights about processing Tamil text using various available tools and libraries which are very limited for this language.

The report is structured as follows. The next section 2 provides related works in Tamil. Methodology is described in Section 3. Subsequently, Section 4 illustrate the design specification of the research project. Section 5 details the approaches that we are experimenting to determine the polarity along with results. Finally, the evaluation and comparison of the proposed model and discussion is presented in Section 6. The Conclusion was provided in section 7.

2 Related Work

2.1 Natural Language Processing

This section aims to introduce the various techniques and tools utilized in the development of natural language processing. According to (Chowdhary; 2020) NLP is focused on the study and development of methods that can be used to analyse and perform different tasks such as speech recognition, machine translation, language text analysis and text summarization. Text manipulation is also widely used in the field of natural language

processing. The studies in the field of natural language processing goes through various steps in the process of translating and processing the text. These include analysing and recognizing the words and their parts-of-speech, the parsing of words, and identifying the patterns and meaning of the text. Sentiment analysis is one of the technique used in NLP to identify the sentiment of the text as positive, negative or neutral. In sentiment analysis, it is also used to find the meaning and pattern of the text and determine its sentiment.

2.2 Sentiment Analysis

In this section, we will talk about the various techniques that are used in sentiment analysis. According to (Van Atteveldt et al.; 2021) in the past, it was done using manual annotators and cookbooks. Although it is more accurate, it can be very time-consuming and expensive to manually code each document.

In the study (Van Atteveldt et al.; 2021) described two types of methods used in sentiment classification. One is dictionary-based and another one is based on machine learning. Although both of these methods are commonly used, machine learning is more likely to perform better among those. This is because of the varying size of the lexicon used in the dictionary-based method. This is why it takes longer to complete the task (Mitra; 2020). To improve the efficiency of sentiment classification in the study conducted by (Mitra; 2020), a rule based approach was implemented. The main features of this approach are speech tagging, tokenization, and stemming. The implementation of this approach involved the use of various machine learning techniques such as KNN, Naïve Bayes, Random Forest, SKlearn SVC and Decision Tree. In the result Random Forest was able to achieve the accuracy of 80% which is higher than other techniques.

However, the result of the experiment (Ahmed et al.; 2021) conducted on the reviews of the amazon fine food shows that the Linear SVC performed well. Compared to (Yadav et al.; 2021) this paper performed several demonstrations for analyzing sentiments that have a large amount of data. They proposed three different algorithms namely, Logistic Regression, Linear SVC and Naïve Bayes which gives an accuracy more than 80%. The paper found that the linear SVC performed better than the other two models. In Contrast, (Yadav et al.; 2021)found that the average AUROC of the linear SVC is less than that of the logistic regression accuracy in their research work. Like (Ahmed et al.; 2021) three methods were then used to train the model namely linear SVC, logistic regression, and Naïve Bayes. The results of the study revealed that the linear SVC performed better than the logistic regression model when it comes to classification accuracy. However, its average AUROC is lower when compared to logistic regression.

In the previous studies, they only considered machine learning to identify the polarity of the text. Most of the time, these applications rely on word frequencies and bag-of-words as their inputs. Sometimes this approach excludes certain aspects of grammar and word order. In order to overcome this limitations, word embedding and deep learning models are often combined to perform the tasks (Van Atteveldt et al.; 2021). Deep learning algorithms which uses multilayers are known to work better on complex process than ML algorithms.

The experiment of (Van Atteveldt et al.; 2021) was conducted to analyse both ML and DL algorithms. They used the data collected by the newspapers and websites to perform sentiment analysis on Dutch economic headlines. The CNN model outperforms the other models in terms of its performance when it comes to extracting the meaning of

words from the headlines. Similarly, in a study (Amulya et al.; 2022) the output revealed that deep learning techniques, such as RNN, CNN, and LSTM, performed better than machine learning algorithms when it came to identifying the polarity of the reviews of movies. Among the techniques RNN reaches the highest of 88% accuracy.

In this subsection, various algorithms that have been proposed to find the sentiment of a text were analysed. The main reason why the results of these studies vary is due to the type of data that it is analyzing and the language that affect the validity of the model. The sentiment analysis itself is strictly considered as a domain-based (Van Atteveldt et al.; 2021). According to the studies presented in this section, the results of DL algorithms are more accurate than those of ML algorithms due to the use of word embeddings as their input.

2.3 Sentiment Analysis Based Deep Learning approach

This section aims to review studies that investigated the use of deep learning techniques in sentiment analysis. The researchers (Zouzou and El Azami; 2021) in their work, presented a new approach to combine the text function model and sentiment-specific word embeddings. In this paper, the authors proposed the method using the word embedding technique known as "gloVe" instead of "Amsterdam embeddings" which was used in the study conducted by (Van Atteveldt et al.; 2021). (Zouzou and El Azami; 2021) focused on developing a text sentiment classification algorithm that combines the CNN-GRU model and the GloVe word embedding. For this experiment, the researchers used 50k movie reviews in the IMDB dataset. After pre-processed the data, firstly the embedded layer was created that takes into account the 320-dimensional vectors of each word. They then trained the data on algorithms such as CNN, GRU and CNN-GRU models. The proposed algorithms GRU and CNN-GRU was mainly composed of two optimization functions known as Adam and Adadelta. During the training phase, these models were able to achieve the higher accuracy of 86.34% for GRU and 82.25% for CNN-GRU.

In a study conducted by (Man and Lin; 2021), they analyze the dataset of the ChnSentiCorp which contains hotel reviews. In the experiment various methods were applied such as SVM, CNN and Att-CNN with word2vec and also CNN with BERT. The experiment result shows that the BERT-CNN outperforms word2vec-CNN and word2vec-SVM by achieving the highest accuracy of 90%. While, (Tan et al.; 2022) developed a robustly-optimized version of the BERT model known as RoBERTa instead of BERT due to its undertrained nature. This paper proposes a hybrid approach that combines the RoBERTa and LSTM. This hybrid model was used to perform various tasks in three different datasets: the US Airline Twitter dataset, the movie review of IMDB, and the Sentiment140 and achieves the F1-score of 0.93, 0.91 and 0.90 respectively. From this result we can able to notice that the models accuracy differs from the dataset's domain.

(Vimali and Murugan; 2021) carried out a different approach by using a bi-directional propagation model called BiLSTM instead of the traditional LSTM. This model adds an advantage of understanding the meaning of the sentence using the bi-directional propagation mechanism. For this they used dataset consist of user reviews in the amazon e-commerce platform. To convert each words to vector they used similar approach to (Zouzou and El Azami; 2021) which is well-known as one-hot coding. After performing various functions the proposed system can able to reach higher accuracy of 90%.

(Kour and Gupta; 2022) also considered Bi-LSTM for their experiment to predict a people mental health condition. For that experiment Bi-LSTM and CNN were combined

to create a hybrid model which uses twitter data that can classify whether the user is depressive or non- depressive. They also compare the hybrid model with other traditional models such as RNN and CNN. It is found that the hybrid model CNN-biLSTM model performed well in terms of accuracy. But, BiLSTM-CNN model which uses Bi-LSTM layer before CNN layer outperforms CNN-BiLSTM model in the experiment conducted by (Pasupa and Seneewong Na Ayutthaya; 2022).

In this subsection, we discussed the various deep learning mechanisms in sentiment analysis. However, we can see that these type of advanced hybrid approaches were tested mostly using English language. The output of these models will vary depending on the domain they are working in and the language they are being processed in. In the next section, we will explore the various techniques that are used to find sentiment in different languages.

2.4 Sentiment Analysis in various languages using deep learning

In this section we will investigate various studies that uses different language text for sentiment analysis.

In a research work (Dashtipour et al.; 2021) proposed deep learning methods that can classify the movie and hotel reviews in Persian language into either positive or negative sentiments. They tested these methods against various ML, such as SVM and logistic regression and with DL such as CNN and LSTM. In order to train the models, they were first convert each Persian word into vectors using fastText word embedding and then the CNN and LSTM layers were used to extract the features. It is noted that the Bi-LSTM has the highest accuracy when it comes to analysing movie review data while the 2D-CNN has the highest accuracy when it comes to analysing hotel data. This shows that the result of the model will vary with different datasets. This also gets proved by the experiment conducted by (Khan et al.; 2021).

In the experiment (Khan et al.; 2021) conducted they found a contrasting result to all the above studies that the machine learning models outperformed the deep learning models. The objective of the study was to classify the sentiment in the text of the Urdu language. They used the UCSA dataset to train the models. They introduced the LSTM and 1D CNN deep learning approaches. This was then compared with some of the traditional ML approaches like (LR, SVM, RF, NB, MLP, AdaBoost). The results revealed that the machine learning model LR achieves the higher accuracy among all. However, the main reason for the poor performance of the deep learning model was due to the lack of vocabulary in the pre-trained fastText model.

However, later they conducted another experiment in that (Khan et al.; 2022) tried a different approach to implement deep learning in word embedding. Instead of fastText embedding, they used word2vec. The goal of this study is to analyze the performance of various word embeddings in Roman and English languages. They use similar approach of (Dashtipour et al.; 2021) where they used Bi-LSTM and 2D-CNN. However, in the experiment (Khan et al.; 2022) conducted instead of using a bidirectional LSTM process, they used two separate LSTM layers which analyse the text in one direction twice. They tested the model using four datasets: RUSA-19, MDPI, UCL, and RUSA. They found that the SVM with word 2vec and CBOW provided better results in analyzing sentiment in Roman Urdu than in English. The combination of BERT, 2 layers of LSTM and SVM performed well in analysing the sentiment of the English text.

In a study (Gowandi et al.; 2021) shows how the hybrid architectures can perform

in analysing Indonesian sentiment in e-commerce reviews. After the training process, the combined models such as LSTM-CNN, CNN-LSTM, GRU-CNN, CNN-GRU, and the standard models such as LSTM, CNN, GRU were implemented. The results of the training process show that the combined models, which are the CNN-LSTM, CNN-GRU, and CNN-GRU models, have an accuracy of 82.71%, 82.69%, and 82.56%, respectively and perform better than the standard models. Similarly, the models such as deep LSTM, GRU, and CNN were developed for Arabic sentiment analysis by (Omara et al.; 2022) for extracting features from character-level representation. The results of the study also revealed that combine the architectures CNN-LSTM can improve the performance of the models with accuracy of 95.14%.

Likewise, (Pasupa and Seneewong Na Ayutthaya; 2022) also proposed a hybrid approach for analyzing sentiment in Thai language. They built a new Thai-SenticNet corpus using LEXiTRON and Volubilis, which is a method for translating text into different languages. They extracted the sentic features from the text using the Thai-senticNet corpus. They then converted the word into vectors using the ASGD Weight-Dropped LSTM model. The hybrid models with the combination of both CNN and bi-LSTM were then evaluated on various datasets, such as ThaiEconTwitter, Wisesight, and ThaiTales. The results of the study revealed that the BiLSTM-CNN hybrid model performed better than the other combination of the models.

Unlike (Pasupa and Seneewong Na Ayutthaya; 2022) where they only experimented with bi-LSTM and CNN, (Senevirathne et al.; 2020) experimented the various sequence models that are used in deep learning, such as RNN, GRU and LSTM. They also explore the use of hybrid models, such as those that are based on hierarchical attention and capsule networks. The researchers analyzed the sentiment dataset of 15059 news comments in Sri Lanka's Sinhala language. They used fastText word embedding and used deep learning approach to train the model. They then compared different combination with CNN, namely CNN-GRU, CNN-LSTM, and CNN-BiLSTM. In the result CNN-BiLSTM achieves a higher accuracy of 62.72% followed by LSTM getting 61.89%. Although, the capsule networks performed better than any other with accuracy of 63.23%. The main reason for the poor performance of the different variants of CNN was mainly due to the lack of data to learn these parameters.

(Miao et al.; 2019) proposes a different approach by combining CNN with bidirectional GRU (CNN-BiGRU) to build a text sentiment analysis model. They collected data and built a corpus of Chinese movie and television reviews. The results of the experiment revealed that the proposed CNN-BiGRU model performed better than the CNN-BiLSTM model in terms of its accuracy and training rate. The experiments also revealed that the output of the proposed CNN-BiGRU model increased by 4.02% and 3.72% compared with the output of the CNN-GRU and GRU models.

Although the accuracy of some of the models was not the same when they were training in English language. But we can note that the hybrid models were able to improve the performance compared to the previous models. This motivated to continue looking into the performance and capabilities of the hybrid model in Tamil language. The field of natural language processing is still in its early stages in Tamil language due to the agglutinative nature of the language (Anbukkarasi and Varadhaganapathy; 2020). So, in the following subsection, our focus will be on the recent work that has been performed in the Tamil language.

2.5 Sentiment analysis in Tamil languages

This section will provide the necessary information to understand and develop effective strategies for performing natural language processing in Tamil.

Its agglutinative nature and grammatical structure make it more challenging (Anbukkarasi and Varadhaganapathy; 2020). (Anbukkarasi and Varadhaganapathy; 2020) compared the performance of the LSTM and BiLSTM networks in terms of their ability to analyze the Tamil tweets. Data pre-processing were done by removing various symbols and punctuation marks. They then trained the word2vec model to convert the words from the Tamil tweets into vectors. Since the characters in the language have special compound characters, they are considered as combined characters instead of individual characters. This makes them different from other NLP tasks. The experiment shows the BiLSTM model and got accuracy of 86.2% that performed better than the LSTM model, which achieves accuracy of 77.2%.

In an experiment (Krishnan et al.; n.d.) developed a method that can classify the sentiments of the users using the tweets in the Tamil and Malayam were proposed. After the pre-prossing of collected data, the models were trained using deep learning techniques. The result shows that LSTM approach were able to achieve a 97.71% accuracy rate on the Tamil dataset and a 97.23% accuracy rate on the Malayam dataset.

(Shanmugavadivel et al.; 2022) takes into account the data set of Tamil and English languages from the Fire 2021 database. To address the class imbalance problem, resampling is performed. The proposed model were analysed with both pre-processed and raw data. The results of this study show that the pre-processing techniques improve the accuracy. This research is focused on developing hybrid deep learning models that are composed of CNN+LSTM, LSTM+CNN, CNN+BiLSTM and BiLSTM+CNN. The result of this study indicate that the CNN+BiLSTM hybrid deep learning model is very effective at analyzing the sentiment generated by Tamil code-mixed data by achieving the accuracy of 66%.

Some other researches were also conducted on Tamil language in terms of classification using natural language processing. For instance, in an experiment (Ramraj et al.; 2020) conducted the Tamil news reports were collected and classified into different topics such as politics, cinema and sports. They compared the methods used by machine learning to extract features from words using TFIDF of words with the deep learning model CNN that uses the Pre-trained Word2Vec as word embedding. The output of the study revealed that models trained with Word2Vec and CNN performed better than the machine learning technique. In addition, due to the presence of a new token in the data, the recall and F1 score from politics tests performed lower than those from sports and cinema.

In this section, we have seen a few studies that were conducted on Tamil language in the field of sentiment analysis.

2.6 Tools and Libraries used for Tamil language

(Arora; 2020) presents the NLP library which is iNLTK. It supports various Indic Languages including English, Hindi, Tamil, Malayalam and Telugu. It also supports code-mixed languages such as Malayalam and English, Tamil and English and so on. Also this iNLTK library provides pre-trained models for data augmentation, word embeddings, Tokenization and son on in various indic Languages. In this proposed paper the author from this iNLTK library used pre-trained language model for classification of the text. The experimented result shows that this helps to achieve more than 95% of accuracy.

In the study (Fernando and Wijayasiriwardhane; 2020) proposed a methodology for identifying religious extremism-based threats in Sri Lanka using social media data from tweets. In this study while preprocessing the several tools were used for tamil and Sinhala language. The preprocessing of Tamil language was done by using existing tools such as IndicNLP and RippleTagger. The RippleTagger is a python library which is used for part of speech (POS) tagging. A part of speech tag, which is also known as POS, is a tool used to identify the parts of a word in a sentence.

(Sarveswaran and Dias; 2021) presents a state-of-the-art, contextual POS tagger called ThamizhiPOSt. They showed how they developed ThamizhiPOSt, a POS tagger for Tamil that uses the Stanza neural-based toolkit. The study then compares ThamizhiPOSt with other Tamil POS taggers. It shows that its accuracy is 93.27% when compared to all the others. It has a score of 93.27%. They have also discussed the various tools and resources that are available for Tamil POS taggers. The proposed TamizhiPOSt can also able to generate the data with POS tags with other features as well such as spell checkers and translations. Also, in a study conducted by (Visuwalingam et al.; 2021), using POS they were able to identify the meaning of the neighboring words.

2.7 Conclusion

These studies revealed the techniques are being used to understand the structure of the language and their meaning. By considering those in this study, we will be presenting the hybrid methods and compare them to find the best model in the analysis of the sentiment in the Tamil Movie reviews. From the above literature review other than CNN-BiLSTM model, The hybrid models such as CNN-LSTM and CNN-BiGRU were also performing better in classifying the sentiment of the text. Although these model's accuracy vary from domain and the language. For example, in the study (Shanmugavadivel et al.; 2022) proposed on Tamil-English code mixed data CNN-BiLSTM performes better than other models. However, in the proposed work of (Miao et al.; 2019) CNN-BiGRU achieves better result than the CNN-BiLSTM.

Proposal	Year	Dataset	Features Extraction and Word Embedding	Model	Accuracy
Shanmugavadivel, K et al.,	2022	Fire 2021 database	TF–IDF score	CNN+BiLSTM	66%
Khan, L et al.,	2022	RUSA	word2vec(CBOW)	CNN-LSTM	84.10%
		RUSA-19			74.80%
		UCL			74%
Khan, L et al	2021	Urdu Corpus for Sentiment analysis (UCSA) dataset	Pre-trained fastText model	2D-CNN	89.76%
				LSTM	75.96%
				LR	81.94%
				SVM	81.47%
Dashtipour, K et al	2021	Persian movie review and hotel review datasets	Pre-trained fastText model	BiLSTM	95.61%
Pasupa, K. and Seneewong Na Ayutthaya, T	2021	ThaiEconTwitter	ASGD Weight-Dropped LSTM model and Thai WordNet	BiLSTM-CNN	74.36%
		ThaiTales			77.07%
		Wisesight			55.21%
Gowandi, T et al.,	2021	Indonesian e-commerce reviews	-	CNN-LSTM	82.71%
				GRU-CNN	82.69%
				CNN-GRU	82.56%
Krishnan, V.G et al.,	2021	Tamil tweets	TF-IDF score	LSTM	87.67%
		Malayam tweets			83.49
Anbukkarasi, S. and Varadhaganapathy, S	2020	Tamil tweets	word2vec	BiLSTM	86.2%
				LSTM	77.2%
Ramraj, S et al.,	2020	Tamil News report	TFIDF and Word2Vec	CNN+wordvec	98%
				SVM + TFIDF	78%
				NB+TFIDF	77%
Senevirathne, L et al.,	2020	15059 news comments in Sinhala language	fastText	CNN-BiLSTM	62.72%
				CNN-LSTM	61.89%
Miao, Y et al.,	2019	Chinese movie and television reviews	Word Vector	BiGRU	83.64%
				CNN-GRU	81.86%
				CNN-BLSTM	83.06%
				CNN-BiGRU	85.58%

Figure 1: Summary of Literature Review

3 Methodology

This section aims to introduce the research methodology used in developing the proposed system. This work followed the KDD framework, which is a stage-wise approach to analyzing large datasets. It aims to find hidden patterns in the data that can be useful in developing effective insights. It also focuses on the process of knowledge discovery to extract useful insights from the data. In addition, it is dedicated to text data mining. The steps involved in KDD process is illustrated in the figure 2. In this research work, hybrid models are created using deep learning algorithms.

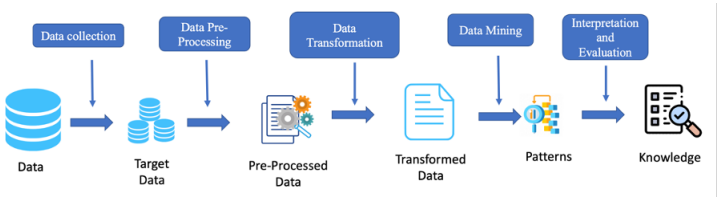


Figure 2: The process of KDD methodology

3.1 Data Collection

The data collected from Kaggle [1] for this project include the ratings and reviews of Tamil movies. All of the reviews are in the Tamil language and are rated from 1 to 5. This collected dataset is publicly available through the Kaggle repository and has no ethical issues.

3.2 Data Pre-Processing

The goal of the pre-processing step is to improve the overall quality of the data. In this step, we will improve the data reliability of raw materials by performing data cleaning. This process involves dealing with various issues such as missing values, duplicate values, and outliers. Unfortunately, in most cases, raw text cannot be analyzed in a quantitative manner. To ensure that data is appropriate for mining, Natural Language Processing performs the task in a tidy manner. Several pre-processing procedures have been implemented in order to improve the quality of text. These include the removal of stop words, special characters, and punctuation using libraries such as nltk, IndicNLP which supports Tamil language.

3.3 Data transformation

After pre-processing, the data was ready for transformation. A rating class was created to classify the data into either positive or negative categories and label them. The ratings were based on the number of reviews that were available online. The lowest score was 1, while the highest was 5. In this step, we transform the multi-class into binary class that is positive as 1 when the rating is above 3 otherwise, it is negative and labelled as 0 (0-Negtive; 1-Positive). The figure 3 shows the distribution of ratings.

[1] `https://www.kaggle.com/datasets/sudalairajkumar/tamil-nlp?select=tamil_movie_reviews_train.csv`

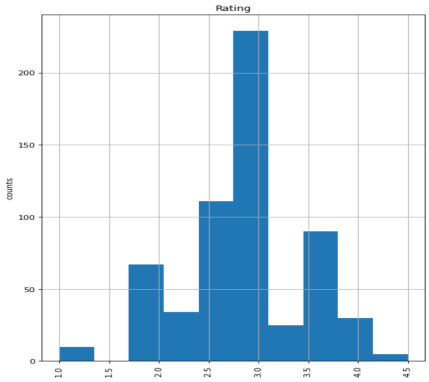


Figure 3: Distributon of Ratings

3.4 Data Mining

The next step involves selecting the appropriate data mining technique for analyzing and extracting patterns from large datasets. This process involves selecting the various modeling techniques that are used to find and extract the patterns from the data. The two main goals of data mining are prediction and description. While prediction is usually performed on a supervised basis, description is usually performed on a more unsupervised model. In this project, deep learning models namely CNN-LSTM, CNN-BiLSTM and CNN-BiGRU will be built to classify the polarity of the Tamil text.

3.5 Data Interpretation and Evaluation

The performance of the model will be evaluated by taking into account the various metrics that are related to its prediction ratios. In addition to the accuracy, multiple metics such as F1 scores and precision will also be used to evaluate the model's performance. The Confusion matrix to evaluate the model performance is shown in the figure 4.

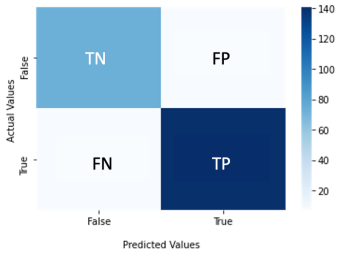


Figure 4: Confusion Matrix

- The TP is a representation of the positive number of samples that it predicts as a positive class.
- The FP is a representation of the negative number of samples that it predicts as a positive class.
- The TN is a representation of the negative number of samples that it predicts as a negative class.

- The FN is a representation of the positive number of samples that it predicts as a negative class.

The following are the major elements of the evaluation matrices:

Accuracy: The accuracy is computed by the ratio of the number of sentiments that it labels correctly to the number of observations.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Precision: The precision metric is used to measure the accuracy of a model's predictions when it comes to analyzing the data collected from various observations. It shows how well the model can predict the true values of the samples. The higher the number of positive samples that the model correctly predicts, the more accurate the model is,

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall: The recall value is a measure of accuracy that is used to analyze the sentiment classification. It helps reduce the number of negative samples that are predicted to be incorrect. This is done by taking into account the ratio of true positive to false negative.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1-Score: F1-score is a measure of harmonic means of the values of precision and recall.

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

4 Design Specification

The Design Specification explains the archietecture of the proposed work shown in figure 5.

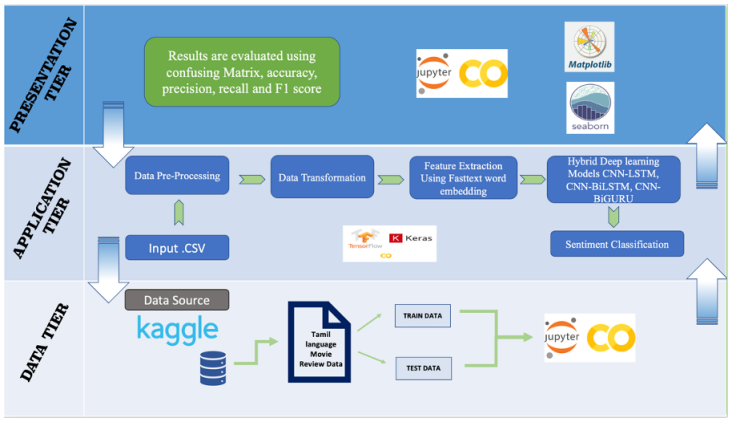


Figure 5: design specification

Tier 3: Data Tier In this stage, the data was collected which is publically available in kaggle. Then collected data was then stores in google drive and then upload into google colabatory notebook to create dataframes that can be used to perform operations.

Tier 2: Application Tier In this phase, the collected raw data was cleaned and transformed for futher application of algorithms. The goal of the project was to analyze sentiments using a clean text pre-processing technique. This process included removing special characters and stopwords, Tokenisation, morphological analysis. This process was carried out using various libraries that supports Tamil, such as nltk, IndicNLP. Then using fasttext word embedding each word was converted into vectors. This input vector was fed into combination of CNN with different deep learning algorithms such as LSTM, BiLSTM and BiGRU. We have used tensorflow and keras libraries to create this three hybrid deep learning models to predict the sentiment.

Tier 1: Presentation Tier In this stage, the result from the previous stage was evaluated using confusion matrix and classification report. Then three models were compared using accuracy, f1-score, recall and precision. This was visualized using libraries such as seaborn and matplotlib.

5 Implementation

5.1 Data

The collected data contains review given by various people. In some instances the raw data contains unwanted symbols or punctuations and noise. So, it is important to remove those symbols and prepare the data for the sentiment classification. Also, in this process we need to remove words that are not useful to the data. Doing so will help minimize the dimensionality of the text and improve its polarity.

5.2 Data Pre-Processing

The following steps were followed to prepare the data by cleaning the noise in the data.

Removal of Punctuation Punctuations are symbols used in writing to explain the meaning of a sentence or a concept. Most social media posts feature a lot of punctuation. Their placement is often random and contradicts the conventions of grammar. This paper aims to construct a noise-free content by removing various punctuation symbols from our data set.

Tokenisation The process of tokenization is used to break down a sentences into various words, which are known as tokens. Generally, these tokens can also be used to represent sentences or paragraphs. In our case, we are breaking each sentence into words using word_tokenise from nltk library (Arora; 2020).

Morphological analysis A morphological analyser is a tool that helps in the retrieval and storage of morphosyntactic and morphophonological information. It is also useful for the study of languages with complex morphological structures. It is a process which is similar to stemming that find the root word. Tamil is a morphologically rich language so this step is considered instead of stemming and lemmatization. For this, we have used 'Indic NLP' library which supports preprocessing of Tamil language (Fernando and Wijayasiriwardhane; 2020).

Removal of stop word One of the most common tasks in the pre-processing of text data is removing unnecessary data. In NLP, these are called stop words. They are

considered to be irrelevant since they are frequently used and do not add any additional value to the context. To save both time and space, these terms are removed. This step is done by using the Tamil stop word provided by TamilNLP.

Padding The next step is padding. When we are working and processing the texts, we need to make sure that the inputs are in the same shape and size. Because, some of the sequences are very long and some are very short. Adding padding is necessary to ensure that the inputs are in the same size at that time. After doing that we got the shape of (601, 1150).

5.3 Feature Extraction

Deep learning is a type of mathematical model that cannot learn directly from text data. It relies on vector data to learn. This means that a word has to be transformed into a vector before they can start learning. A word embedding can be used to extract differences and similarities between words. It can also generate relations between them. This is done by converting each word into a numerical vector. This process is called word embedding and this process was done by using pre-trained fasttext word embedding model. The fastText-Skip Gram model was created using over 1.7 million Tamil documents and over 21 million words. It was then exported to a vector file with a dimension of 300. This fasttext embedding model is trained on the cleaned data and vectors for each word was created.

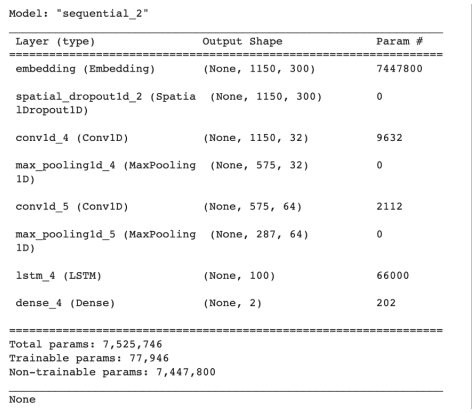
5.4 Learning Models

Studies suggest that CNN is the best choice when it comes to extracting complex data features from textual data. RNNs works better in classification by capturing information from the data (Kour and Gupta; 2022). After extracting features using the Convolutional neural network the input were fed into different deep learning algorithms such as LSTM, BiLSTM and BiGRU to classify the polarity of the text as positive or negative. Hence, in this experiment three hybrid models namely CNN-LSTM, CNN-BiLSTM and CNN-BiGRU were developed and the result were evaluated to analyse the best performing model.

5.4.1 CNN-LSTM

Long-shot memory is a type of RNN that can learn complex long-term dependencies. It was originally developed to address the issue of gradient vanishing, which prevented the RNN from using all input sequences. LSTM allows us to perform various NLP tasks such as machine translation and recognition of handwritten notes. The LSTM architecture features three gates: the input, the output, and the forget. This gates controls the flow of the data. In this study archietecture of CNN to extract the features and LSTM to understand the meaning of the context for classification were developed. The Figure 6 shows the summary of the CNN-LSTM.

The CNN layer takes the local data features and automatically extract them from the input vectors. We used two Convolutional layer with the filter size of 32 and 64. Then the max-pooling with 2 pool size and relu activation function were applied. Then the LSTM architecture then receives the extracted data as input. It consist of 100 hidden neurons with dropout and recurrent droupout rate of 0.2 each. Then the output layer is the dense layer with 2 neurons and has the activation function of softmax.

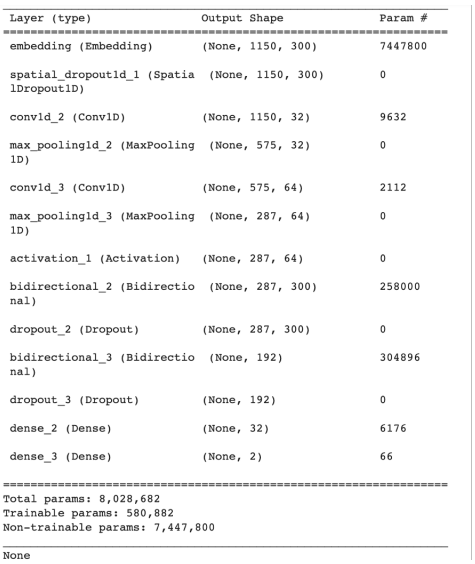


```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1150, 300)         7447800

 spatial_dropout1d_2 (Spatia  (None, 1150, 300)        0
 lDropout1D)

 conv1d_4 (Conv1D)           (None, 1150, 32)          9632

 max_pooling1d_4 (MaxPooling  (None, 575, 32)          0
 1D)

 conv1d_5 (Conv1D)           (None, 575, 64)           2112

 max_pooling1d_5 (MaxPooling  (None, 287, 64)          0
 1D)

 lstm_4 (LSTM)               (None, 100)               66000

 dense_4 (Dense)             (None, 2)                 202

=================================================================
Total params: 7,525,746
Trainable params: 77,946
Non-trainable params: 7,447,800
_________________________________________________________________
None
```

Figure 6: Summary of CNN-LSTM Model

5.4.2 CNN-BiLSTM

Although the LSTM can consider the past, it does not take into account the future context of the input. This is why BiLSTM was developed to allow us to consider both the present and the past. In order to overcome the issue of gradient vanishing, BiLSTM will be used which consider both the past and the future context of the text (Pasupa and Seneewong Na Ayutthaya; 2022). It composed of two hidden layers that work in backward and the forward direction. This forward and backward hidden layers analyse the input in ascending and descending order respectively.



```
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1150, 300)         7447800

 spatial_dropout1d_1 (Spatia  (None, 1150, 300)        0
 lDropout1D)

 conv1d_2 (Conv1D)           (None, 1150, 32)          9632

 max_pooling1d_2 (MaxPooling  (None, 575, 32)          0
 1D)

 conv1d_3 (Conv1D)           (None, 575, 64)           2112

 max_pooling1d_3 (MaxPooling  (None, 287, 64)          0
 1D)

 activation_1 (Activation)   (None, 287, 64)           0

 bidirectional_2 (Bidirectio  (None, 287, 300)         258000
 nal)

 dropout_2 (Dropout)         (None, 287, 300)          0

 bidirectional_3 (Bidirectio  (None, 192)              304896
 nal)

 dropout_3 (Dropout)         (None, 192)               0

 dense_2 (Dense)             (None, 32)                6176

 dense_3 (Dense)             (None, 2)                 66

=================================================================
Total params: 8,028,682
Trainable params: 580,882
Non-trainable params: 7,447,800
_________________________________________________________________
None
```

Figure 7: Summary of CNN-BiLSTM Model

The combination of CNN and BiLSTM model summary is shown in the figure 7. It will allow the model to extract the relevant information from the text and classifying its polarity. The features from both branches are then fed into the last layer, which generates the target predictions. Firstly, the vectors given by fastText word embedding were fed as an input into the CNN layer. In the CNN, the input vectors are fed to 1-D convolutional layer. This layer use 32 filters to perform deep learning on the low-level implicit features

of the raw well logs. They then reduce the dimension of the feature maps by performing a max-pooling operation with pool size of 2. In the second CNN layer filter size and kernel size are provided with the value of 64 and 1 respectively. ReLU is then used to perform the activation function of this layer. A flatten layer is then used to change the dimension of the feature map. In BiLSTM, two-layer of BiLSTM were used that captures the contextual information from well logs. It learns the knowledge from the previous term of log data, and it also learns from the succeeding term of log sequence. The first layer of BiLSTM is a hidden layer that has 150 memory cell units. A dropout operation with rate of 0.3 and 0.2 is performed to prevent from overfitting. The second layer of the Bi-LSTM network is composed of 96 neurons. This layer is followed by an activation function called ReLu. The last dense layer contains 2 neuron and SoftMax activation function is used for the output layer.

5.4.3 CNN-BiGRU

The GRU(gated recurrent neural network) is a type of neural network that is designed to provide a secure and resilient memory network. It is similar to the long- and short-term memory networks. In this proposed work, BiGRU is used to extract the contextual sequence information features of a text (Miao et al.; 2019). It takes into account the direction of the positive input sequence and the reverse sequence direction. When performing feature extraction on the input sequence, the two Gated Recurrent Neural Networks do not share the same state. The GRU state transition rules follow the same procedure when performing feature extraction on the same states. However, when the output results of the two GRUs are split into the BiGRU layer's output, the result of the extraction is considered as the output of the entire layer. The CNN-BiGRU model is a framework that combines the capabilities of CNN and BiGRU to extract text features and sentence representations shown in 8. It helps in solving the long-term dependence issue by adding output gates, input gates, and forget gates.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 1150, 300)         7447800

spatial_dropout1d_3 (Spatia  (None, 1150, 300)         0
lDropout1D)

conv1d_6 (Conv1D)            (None, 1150, 64)          19264

max_pooling1d_6 (MaxPooling  (None, 575, 64)           0
1D)

activation_2 (Activation)    (None, 575, 64)           0

spatial_dropout1d_4 (Spatia  (None, 575, 64)           0
lDropout1D)

bidirectional_4 (Bidirectio  (None, 150)               63450
nal)

dropout_4 (Dropout)          (None, 150)               0

dense_5 (Dense)              (None, 2)                 302

=================================================================
Total params: 7,530,816
Trainable params: 83,016
Non-trainable params: 7,447,800
_________________________________________________________________
None
```

Figure 8: Summary of CNN-BiGRU Model

The first layer is the embedding layer which is the input that coverts words into numerical representation. Then the input were fed into the CNN layer is to extract the local static feature from a text. BiGRU then performs the extraction of the sequence semantic information. The archietecture of CNN consist of layers such as two convolutional layers, two max-pooling layer and activation function. The convolution layers hold the filter

size of 32 and 64 with activation function of relu. The ouput were fed into the spatial droput1D of 0.2. The BiGRU architecture consist of several layers such as BiGRU layer, dropout layer and fully connected layer. The output data were given to the BiGRU layer that consist of 75 neurons. Then the droupout layer were applied to avoid the issue of overfitting. The fully connected layer were used with the dense of 64 and 32 neurons with relu activation function. In the output layer dense with 2 neurons and softmax activation function were used to classify the features.

6 Evaluation

6.1 Experiment 1/ CNN-LSTM

The model is evaluated after tuning its parameters. The epoch size were given as 20. As the size of the data is relatively small the batch size is given as 32. In addition, to prevent the model from overfitting we have used the early stopping with patience as 3 and loss as monitoring parameter. The Early stopping method allows you to stop training a model once its performance stops improving. This method can be used to specify a large number of training epochs.

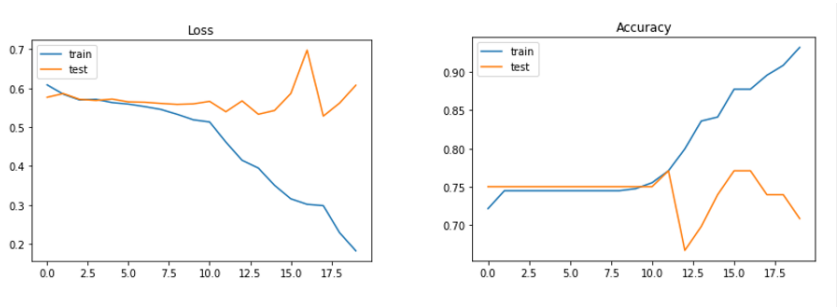


Figure 9: Graph of Loss and Accuracy for train and test data using CNN-LSTM

The CNN-LSTM model performed well and achieves the training accuracy of 94.17%. Although, when it comes to testing accuracy CNN-LSTM can only able to achieve about 77%. From the figure 9 we can observe that the loss value for the train data get decreasing gradually which increase the accuracy of the training and the loss for testing is almost stable.

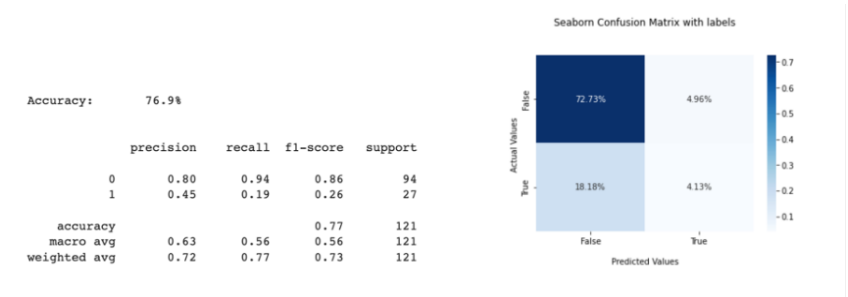


Figure 10: Classification Report and Confusion Matrix for CNN-LSTM

From the figure 10 we can see the classification report for CNN-LSTM. It shows the values of evaluation matrix such as precision, recall and f1-score. Also, from the confusion matrix we can find the summary of correctly and wrongly predicted values. The model predicted 72.73% as negative labels correctly and performed poorly in predicting positive labels. This might be because of the unbalanced nature of the labels in the dataset.

6.2 Experiment 2/ CNN-BILSTM:

Initially, the CNN-BiLSTM model is developed with default parameters. To avoid the issues related to vanishing gradients, a ReLu activation function is used. A dropout layer is also added in the model to prevent the model from overfitting. Then model is evaluated through the tuning its various hyperparameters, such as the optimizer, loss function, metrics and early stopping. For this model the best hyperparameters are Adam for optimization, categorical cross-enstropy for loss function, and early stopping with loss for monitor the performance of the model. The CNN-BiLSTM model achieved an accuracy of 80.2%.

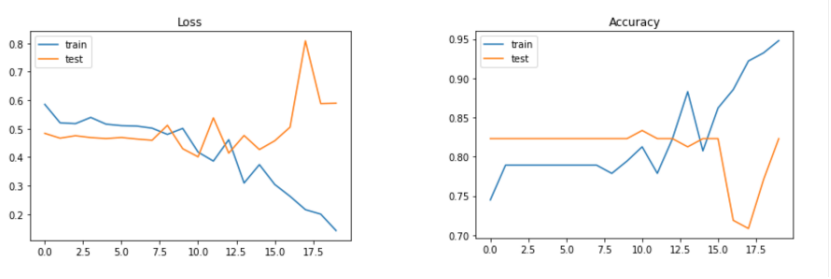
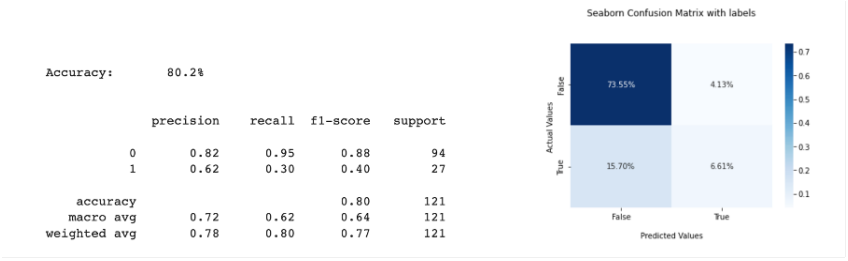


Figure 11: Graph of Loss and Accuracy for train and test data using CNN-BiLSTM

It is noted that the training accuracy and the validation accuracy of a model remains constant for almost 7 epoch. Then the training accuracy gets gradually increasing and achieve the highest accuracy as 94.17%. However, while testing data is compared with training data, its accuracy drops below the model accuracy that able to achieve 80.2%. This can be seen in the graph illustrated in the figure above. The figure 11 also shows the loss function graph for training and testing data. It shows that the loss function for test was almost stable with little fluctuation and after some epochs it get increased slightly, while the train loss drops down rapidly after some epochs . We can note that this is the reason for the increased training accuracy. The confusion matrix is also used to evaluate the result of the model which is shown in the figure below.



Accuracy: 80.2%

	precision	recall	f1-score	support
0	0.82	0.95	0.88	94
1	0.62	0.30	0.40	27
accuracy			0.80	121
macro avg	0.72	0.62	0.64	121
weighted avg	0.78	0.80	0.77	121

Figure 12: Classification Report and Confusion Matrix for CNN-BiLSTM

The performance of the hybrid CNN-BiLSTM model was evaluated using confusion matrix and classification report which were shown in the figure 12. The report shows that the model has an overall accuracy of 0.80. The average precision predicted by the model for both positive and negative classes is 0.72. The recall score for the model is 0.62 percent. The macro average for f1 score is 0.64 tells that the model is good in predicting the negative labels. Whereas, its performed less in predicting the positive labels.

6.3 Experiment 3/ CNN-BiGRU:

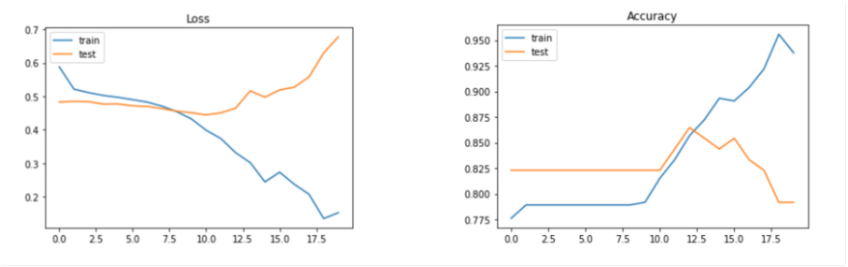


Figure 13: Graph of Loss and Accuracy for train and test data using CNN-BiGRU

The hyperparameter for this model was provided as adam for optimizer, categorica_crossentropy and metrics where choosen as accuracy to get the accuracy value in the output. And the epoch size were set as 20 with addition of early stopping. For this model also we kept the batch size as 32 to make the model performance more accurate. The lower the batch size better the performance of the model will be (Gowandi et al.; 2021).

The graphs in the figure 13 indicate that the model performed good in training. The model's accuracy on testing and training data was observed to be roughly same in the starting period and then the training accuracy gets increased rapidly while the testing accuracy becomes stable which is lower than the training. When training the model it gets an accuracy of 95% but the accuracy was lower in the testing which is 76%. A classification matrix is then generated to evaluate the model's observation.

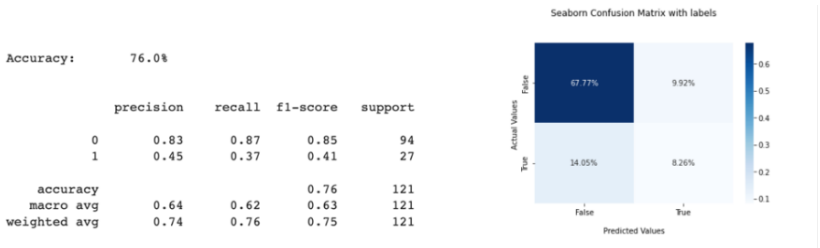


Figure 14: Classification Report and Confusion Matrix for CNN-BiGRU

The figure 14 confusion matrix shows that the overall accuracy of the CNN-BiGRU reaches 79%. The classification summary shows that the model performed better in predicting negative labels than the positive labels which can be seen from the f1-Score of 0.83 for negative labels and 0.45 for positive. This may be due to the unbalanced labels in the data as well. The confusion matrix also indicates this by having 67.77% in predicting

negative labels as negative and only 8.26% for predicting positive labels correctly. The overall precision, recall, f1-score was calculated as 0.64, 0.62 and 0.63 respectively.

6.4 Comparing the Hybrid Deep learning Models

Table 1: Comparsion Table

Model	Training Accuracy	Testing Accuracy	Precision	Recall	F1-score
CNN-LSTM	94.17%	77%	0.63	0.56	0.56
CNN-BiLSTM	94.17%	**80.2%**	**0.72**	**0.62**	**0.64**
CNN-BiGRU	95%	76%	0.64	0.62	0.63

6.5 Discussion

In this section, critical analysis regarding the findings of the research will be carried out. The aim of the project was to build hydrid deep learning model that can provide an accurate and efficient outcome by analyzing sentiment of the Tamil movie reviews. After conducting the literature review and a review of the various aspects of the project, the current limitations and gaps was discovered. There are very limited resources were available for making the analysis on various regional languages. Even though the resources are limited recent studies have shown more interests in bringing new advancement by making use of the available data. Most of the research done in the Tamil language were based on the traditional machine learning or the baseline deep learning model only. The hybrid model of combing two models by using word embedding technique were less explored. For this study, three hybrid deep learning models were implemented and have been compared. All the three models performed reasonably well in the pre-processed data. Among them, CNN-BiLSTM performed better by achieving the highest accuracy of 80.2% while the CNN with LSTM and BiGRU achieves the overall accuracy of 77% and 76% respectively. All the three model were able to achieve the accuracy of above 90% while training. The reason for the drop during testing might be the smaller size of dataset. Moreover, the prediction labels were unbalanced that has more negative reviews than the positive reviews. This might cause the misclassification during the testing. Although sampling techniques are commonly used to balance the variables which was not performed in this study which might divert from the research objective. Because, the objective of the research is to develop three hybrid deep learning model and compare them to identify the best performing model for the Tamil reviews sentiment classification. Hence, by taking the research question into account, the experiments revealed that the CNN-BiLSTM is a better model to classify the sentiment in the Tamil text.

7 Conclusion and Future Work

The main aim of this research work is to analyse the best performing model for sentiment classification in Tamil language by evaluating and comparing them. For this study, Tamil movie reviews data were collected from Kaggle [2]. Then pre-processing techniques are also

[2] https://www.kaggle.com/datasets/sudalairajkumar/tamil-nlp?select=tamil_movie_reviews_train.csv

examined to improve the efficiency of the system. These include the use of punctuation removal, tokenisation, morphological analysis, stopword removal and padding. The data was pre-processed using various tools and libraries available for the Tamil language such as nltk, indicnlp. Then embedding layer was created using fasttext word embedding which is pre-trained on the Tamil text and it provide 300 dimensional vectors. The cleaned data were then used to train the models namely CNN-LSTM, CNN-BiLSTM, and CNN-BiGRU. Each model were trained with hyperparameters like optimizer, loss function and also by increasing the epoch size and constant batch size of 32. The models were evaluated with the use of classification report and confusion matrix. All the models got an accuracy of over 90% in the training process but this got drop down in the testing. Eventhough, in the training process CNN-LSTM and CNN-BiLSTM got the same accuracy of 94.17%, while testing the model on test data the CNN-BiLSTM outperforms the CNN-LSTM. In the result, the proposed CNN-BiLSTM achieves the overall higher accuracy of 80.2% followed by CNN-LSTM with 77% and CNN-BiGRU with 76%. Also, CNN-BiLSTM got the highest f1-score of 0.64, where CNN-LSTM and CNN-BiGRU got 0.56 and 0.63 respectively.

In the future work, in order to improve the performance of models, a large dataset will be used. This will allow them to perform better in terms of their results with balanced data. In addition, with the knowledge gain from this project aspect based sentiment analysis will be carried out to classify the Tamil text and identifying the polarity for each aspect in the text.

8 Acknowledgement

I would like to convey my sincere gratitude to my supervisor, Taimur Hafeez for his guidance and support throughout this three month. His valuable feedbacks helped me complete my research project. Also, I would like to thank my family and friends to encourage me during the research process.

References

Ahmed, H. M., Javed Awan, M., Khan, N. S., Yasin, A. and Faisal Shehzad, H. M. (2021). Sentiment analysis of online food reviews using big data analytics, *Hafiz Muhammad Ahmed, Mazhar Javed Awan, Nabeel Sabir Khan, Awais Yasin, Hafiz Muhammad Faisal Shehzad (2021) Sentiment Analysis of Online Food Reviews using Big Data Analytics. Elementary Education Online* **20**(2): 827–836.

Amulya, K., Swathi, S., Kamakshi, P. and Bhavani, Y. (2022). Sentiment analysis on imdb movie reviews using machine learning and deep learning algorithms, *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, pp. 814–819.

Anbukkarasi, S. and Varadhaganapathy, S. (2020). Analyzing sentiment in tamil tweets using deep neural network, *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, pp. 449–453.

Arora, G. (2020). inltk: Natural language toolkit for indic languages, *arXiv preprint arXiv:2009.12534* .

Chowdhary, K. (2020). Natural language processing, *Fundamentals of artificial intelligence* pp. 603–649.

Dashtipour, K., Gogate, M., Adeel, A., Larijani, H. and Hussain, A. (2021). Sentiment analysis of persian movie reviews using deep learning, *Entropy* **23**(5): 596.

Fernando, A. and Wijayasiriwardhane, T. K. (2020). Identifying religious extremism-based threats in srilanka using bilingual social media intelligence, *2020 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, IEEE, pp. 103–110.

Gowandi, T., Murfi, H. and Nurrohmah, S. (2021). Performance analysis of hybrid architectures of deep learning for indonesian sentiment analysis, *International Conference on Soft Computing in Data Science*, Springer, pp. 18–27.

Khan, L., Amjad, A., Afaq, K. M. and Chang, H.-T. (2022). Deep sentiment analysis using cnn-lstm architecture of english and roman urdu text shared in social media, *Applied Sciences* **12**(5): 2694.

Khan, L., Amjad, A., Ashraf, N., Chang, H.-T. and Gelbukh, A. (2021). Urdu sentiment analysis with deep learning methods, *IEEE Access* **9**: 97803–97812.

Kour, H. and Gupta, M. K. (2022). An hybrid deep learning approach for depression prediction from user tweets using feature-rich cnn and bi-directional lstm, *Multimedia Tools and Applications* pp. 1–37.

Krishnan, V. G., Rao, P. V., Deepa, J. and Divya, V. (n.d.). Twitter sentiment analysis using ensemble classifiers on tamil and malayalam languages.

Man, R. and Lin, K. (2021). Sentiment analysis algorithm based on bert and convolutional neural network, *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, IEEE, pp. 769–772.

Miao, Y., Ji, Y. and Peng, E. (2019). Application of cnn-bigru model in chinese short text sentiment analysis, *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pp. 510–514.

Mitra, A. (2020). Sentiment analysis using machine learning approaches (lexicon based on movie review dataset), *Journal of Ubiquitous Computing and Communication Technologies (UCCT)* **2**(03): 145–152.

Omara, E., Mosa, M. and Ismail, N. (2022). Applying recurrent networks for arabic sentiment analysis, *Menoufia Journal of Electronic Engineering Research* **31**(1): 21–28.

Pasupa, K. and Seneewong Na Ayutthaya, T. (2022). Hybrid deep learning models for thai sentiment analysis, *Cognitive Computation* **14**(1): 167–193.

Ramanathan, V., Meyyappan, T. and Thamarai, S. (2021). Sentiment analysis: An approach for analysing tamil movie reviews using tamil tweets, *Recent Advances in Mathematical Research and Computer Science* **3**: 28–39.

Ramraj, S., Arthi, R., Murugan, S. and Julie, M. (2020). Topic categorization of tamil news articles using pretrained word2vec embeddings with convolutional neural network, *2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)*, IEEE, pp. 1–4.

Sarveswaran, K. and Dias, G. (2021). Building a part of speech tagger for the tamil language, *2021 International Conference on Asian Language Processing (IALP)*, IEEE, pp. 286–291.

Senevirathne, L., Demotte, P., Karunanayake, B., Munasinghe, U. and Ranathunga, S. (2020). Sentiment analysis for sinhala language using deep learning techniques, *arXiv preprint arXiv:2011.07280* .

Shanmugavadivel, K., Sampath, S. H., Nandhakumar, P., Mahalingam, P., Subramanian, M., Kumaresan, P. K. and Priyadharshini, R. (2022). An analysis of machine learning models for sentiment analysis of tamil code-mixed data, *Computer Speech & Language* p. 101407.

Tan, K. L., Lee, C. P., Anbananthen, K. S. M. and Lim, K. M. (2022). Roberta-lstm: A hybrid model for sentiment analysis with transformer and recurrent neural network, *IEEE Access* **10**: 21517–21525.

Van Atteveldt, W., Van der Velden, M. A. and Boukes, M. (2021). The validity of sentiment analysis: Comparing manual annotation, crowd-coding, dictionary approaches, and machine learning algorithms, *Communication Methods and Measures* **15**(2): 121–140.

Vimali, J. and Murugan, S. (2021). A text based sentiment analysis model using bi-directional lstm networks, *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, pp. 1652–1658.

Visuwalingam, H., Sakuntharaj, R. and Ragel, R. G. (2021). Part of speech tagging for tamil language using deep learning, *2021 IEEE 16th International Conference on Industrial and Information Systems (ICIIS)*, IEEE, pp. 157–161.

Yadav, N., Kudale, O., Rao, A., Gupta, S. and Shitole, A. (2021). Twitter sentiment analysis using supervised machine learning, *Intelligent Data Communication Technologies and Internet of Things*, Springer, pp. 631–642.

Zouzou, A. and El Azami, I. (2021). Text sentiment analysis with cnn & gru model using glove, *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)*, IEEE, pp. 1–5.