A Deep Learning Approach to Movie Review Sentiment Classification

1st Sudipta Roy Chowdhury
dept. of Computer Science and Engineering
Rangamati Science and Technology University
Jhagrabil, Rangamati Sadar, Rangamati
sudiptarmstu@gmail.com

***Abstract*—This paper presents a comparative study on sentiment analysis using four different deep learning architectures: Long Short-Term Memory (LSTM), CNN-LSTM, Bidirectional Gated Recurrent Unit (BiGRU), and DistilBERT. The models were evaluated on the IMDb movie review dataset, with preprocessing steps including tokenization and sequence padding to ensure consistency. The LSTM, CNN-LSTM, and BiGRU models were implemented using TensorFlow/Keras, while DistilBERT was employed via the Hugging Face Transformers pipeline. Experimental results show that each model achieves competitive accuracy on a subset of the test data, with transformer-based DistilBERT offering strong performance with minimal preprocessing. A detailed discussion of architecture design, training parameters, and evaluation metrics is provided. The paper concludes with insights on the trade-offs between model complexity, training time, and accuracy, as well as recommendations for future work to further optimize model performance.**

Keywords—Sentiment Analysis, Deep Learning, LSTM, CNN-LSTM, BiGRU, DistilBERT, Natural Language Processing, IMDb Dataset.

I. INTRODUCTION

Sentiment classification of movie reviews has been an active research area for over a decade. With the rapid progress in deep learning, researchers have moved beyond traditional machine learning methods to employ sophisticated neural architectures that can capture complex linguistic patterns and contextual nuances in text. Recent studies leverage large-scale pre-trained language models and hybrid architectures, enabling more accurate and fine-grained sentiment predictions.

A. Problem Definition:

The core problem addressed in this study is the binary classification of text sentiment, specifically determining whether movie reviews from the IMDb dataset express positive or negative opinions. Traditional methods often fall short in capturing the nuanced context of language, which poses a significant challenge in automated sentiment classification.

B. Motivation:

Deep learning has transformed NLP by enabling models to learn intricate patterns in data without the need for manual feature engineering. The advent of architectures like LSTM, CNN-LSTM, BiGRU, and transformer-based models such as DistilBERT provides an opportunity to explore and compare these techniques on a common benchmark dataset. This comparative study is motivated by the desire to identify the trade-offs between model complexity, computational efficiency, and classification performance, and to offer insights into which architectures are best suited for practical sentiment analysis tasks.

C. Significance:

The significance of this work lies in its potential to enhance real-world applications of sentiment analysis. Accurate sentiment classification can drive improved customer insights, inform business decisions, and enable more responsive and adaptive systems in industries such as e-commerce, social media, and public relations. By evaluating multiple architectures, this study contributes to the understanding of how deep learning models can be optimized for specific tasks, paving the way for more robust and scalable NLP systems.

II. RELATED WORK

Khan et al. [1] presents a comprehensive approach to sentiment analysis using multiple deep learning architectures on movie reviews. The authors evaluate models including Long Short-Term Memory (LSTM), Convolutional Neural Network–LSTM (CNN-LSTM), Bidirectional Encoder Representations from Transformers (BERT), and XLNet. Using a novel Rotten Tomatoes dataset, they emphasize extensive preprocessing and hyperparameter optimization. Notably, the study finds that transformer-based models—particularly XLNet—can capture long-range dependencies by considering all possible token permutations, resulting in superior performance compared to more traditional recurrent architectures.

Gibson et al. [2] developed both binary and fine-grained sentiment classification by combining the strengths of pre-trained transformers with recurrent neural networks. Specifically, the authors fine-tune a BERT model and further enhance it with a Bidirectional LSTM (BiLSTM) layer. This hybrid approach leverages BERT's powerful contextual embeddings while allowing the BiLSTM to capture sequential dynamics and long-term dependencies, crucial for detecting subtle sentiment variations. Experimental results indicate that this fine-tuning strategy not only improves overall accuracy on benchmark datasets (e.g., IMDb) but also significantly boosts performance on multi-class (fine-grained) sentiment tasks.

Mohamed et al. [3] offers a broad quantitative analysis of more than 100 deep learning–based sentiment classification

approaches across multiple datasets. The work categorizes performance-affecting factors into three groups: data preparation, feature representation, and classifier design. It highlights that differences in word embedding techniques, network architectures (e.g., CNN, RNN, hybrid models), and training strategies significantly influence accuracy. This survey not only benchmarks state-of-the-art methods but also provides insights into why transformer models are increasingly popular, due to their ability to harness contextual information and reduce dependency on manual feature engineering.

Minghao et al. [4] confirms the efficacy of deep learning in sentiment classification by implementing models that range from simple convolutional neural networks to more advanced transformer-based architectures. The work emphasizes practical aspects such as computational efficiency and the importance of robust preprocessing pipelines. By comparing various deep learning techniques on the IMDb dataset, the authors illustrate that while simpler models can achieve reasonable performance, the integration of advanced architectures like BERT and XLNet is essential for capturing intricate language features and achieving state-of-the-art results.

Peter et al. [5] presents a scalable frameworks combining positional embeddings with pretrained word vectors, such as GloVe, have been developed to improve sentiment analysis in online movie reviews. One such model, PEW-MCAB, integrates a multichannel CNN with an attention-based BiLSTM, achieving accuracy rates of up to 90.3% on the IMDB dataset.

This paper examines online theater reviews as the research subject. It obtains review data for “Beyond the Clouds” using web scraping and employs methods such as word cloud analysis, word frequency analysis, positive-negative sentiment analysis, and semantic network analysis to explore the sentiment information contained within. The results indicate that moviegoers strongly identify with and appreciate the real people and true events depicted in such films, with user comments primarily reflecting positive emotions [6].

Tis study offers sentiment-based movie recommendation system research employs techniques such as natural language processing and hybrid models with the goal of increasing user satisfaction. To this purpose, the integration of advanced machine learning algorithms such as cosine similarity, support vector machine, and Naive Bayes improves recommendation systems with sentiment analysis. Cosine similarity improves movie recommendations by recognizing minor user preferences, while support vector machines and Naive Bayes enhance sentiment analysis by offering a nuanced interpretation of textual attitudes [7].

This review delves into the intricate landscape of sentiment analysis, exploring its significance, challenges, and evolving methodologies.This study examines crucial aspects such as dataset selection, algorithm choice, language considerations, and emerging sentiment tasks. The suitability of established datasets (e.g., IMDB Movie Reviews, Twitter Sentiment Dataset) and deep learning techniques (e.g., BERT) for sentiment analysis is examined [8].

This paper provides a comprehensive review of the application of ANNs and RNNs in sentiment analysis specifically focused on movie reviews. The review begins by discussing the fundamental concepts and principles of ANNs and RNNs, highlighting their respective architectures and learning mechanisms. It then delves into the various techniques and strategies employed in sentiment analysis using ANNs and RNNs. These techniques include feature extraction, text representation, and model training, which are critical components in achieving accurate sentiment classification [9].

The three Jurassic World films are subjected to a machine learning topic modeling approach (Latent Dirichlet Analysis, or LDA) in this work. A dataset of reviews collected from the IMDB website is used for the analysis. Each of the three films has a good and a negative subset of the reviews, which are divided into six datasets. Word clouds of the most important terms are used to illustrate the subject modeling results. The word cloud analysis that follows illustrates the variety of themes covered in reviews as well as the type of ambiguity that frequently makes vocabulary-based sentiment analysis more difficult [10].

III. METHODOLOGY

A. Load IMDb Dataset:

The dataset is loaded using the datasets library. It contains 50,000 movie reviews, equally split into positive and negative sentiments. The dataset is explicitly re-downloaded to ensure the latest version is used. The dataset is balanced, ensuring that there is no class imbalance, which could affect model training. The text data includes a mix of different lengths and writing styles, making it a robust dataset for sentiment analysis.

B. Preprocessing:

a) Tokenize Reviews (Keras Tokenizer): A Tokenizer is initialized with a vocabulary size of 10,000. It converts text reviews into sequences of integer tokens. Any word not in the vocabulary is replaced with a special <OOV> (Out-Of-Vocabulary) token to handle unseen words. Tokenization helps in converting text into numerical format, which is required for deep learning models. The tokenizer is fit on the training data to ensure consistency.

- Example transformation:
  Input: "The movie was fantastic!"
  Tokenized: [12, 24, 5, 789]

b) Pad Sequences (Max Length = 500): Using pad_sequences(), sequences are padded to a fixed length of 500 to ensure uniform input dimensions. Shorter sequences are padded with zeros, and longer sequences are truncated. Padding improves batch processing efficiency for deep learning models. The maximum length of 500 is chosen based on the average review length in the dataset.

- Example:
  Before padding: [12, 24, 5, 789]
  After padding: [0, 0, ..., 12, 24, 5, 789]

c) Prepare Labels (Convert to NumPy Array): Sentiment labels (0 = Negative, 1 = Positive) are converted into a NumPy array. Using numpy.array() for faster

computation and compatibility with TensorFlow models. Enables efficient training and evaluation of deep learning models. Ensures data format consistency across different training pipelines.

The proposed approach can be seen in "Fig. 1".

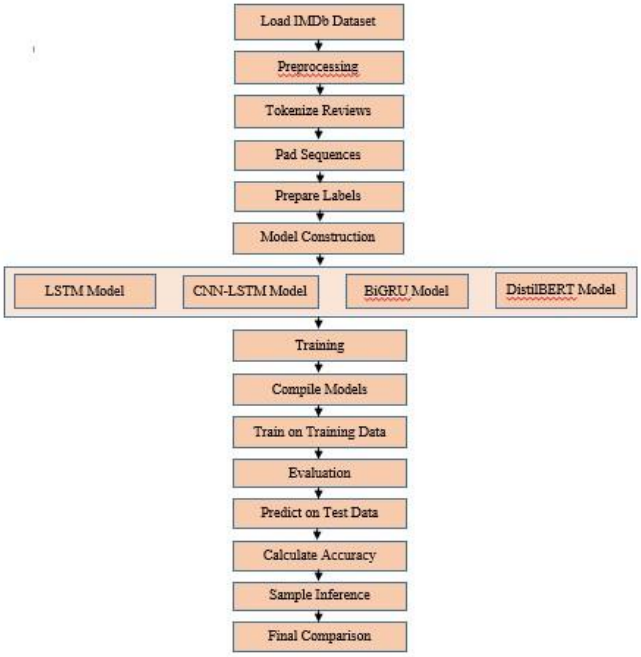

Fig. 1. Proposed Architecture of Movie Review Sentiment Classification

C. Model Construction:

Each deep learning model follows a different approach to process text for sentiment classification.

a) LSTM Model (Embedding → LSTM x2 → Dense): It converts integer sequences into dense vector representations of words. Two stacked Long Short-Term Memory (LSTM) layers capture sequential patterns in text. Fully connected layer with ReLU activation. Sigmoid activation for binary classification. It captures long-term dependencies. It handles sequential text data efficiently. Suitable for text data with temporal dependencies.

```
Training LSTM model...
Epoch 1/3
196/196 ━━━━━━━━━━━━━━━━━━━━ 18s 63ms/step - accuracy: 0.5036 - loss: 0.6934 - val_accuracy: 0.9784 - val_loss: 0.6318
Epoch 2/3
196/196 ━━━━━━━━━━━━━━━━━━━━ 17s 61ms/step - accuracy: 0.5138 - loss: 0.6870 - val_accuracy: 0.0558 - val_loss: 0.6878
Epoch 3/3
196/196 ━━━━━━━━━━━━━━━━━━━━ 20s 59ms/step - accuracy: 0.5339 - loss: 0.6645 - val_accuracy: 0.7820 - val_loss: 0.6782
```

Fig. 2. Training LSTM Model

b) CNN-LSTM Model (Embedding → Conv1D → LSTM → Dense): It converts tokenized sequences into dense vectors. It extracts local features from text sequences. Captures sequential dependencies after feature extraction. Fully connected layer for feature processing. Sigmoid activation for classification. CNN captures local text features before feeding them to LSTM. It is Suitable for extracting key phrases or patterns and combines local feature extraction with sequential pattern recognition.

```
Training CNN-LSTM model...
Epoch 1/3
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/embedding.py:90: UserWarning: Argument `input_length` is de
  warnings.warn(
196/196 ━━━━━━━━━━━━━━━━━━━━ 10s 34ms/step - accuracy: 0.5081 - loss: 0.6924 - val_accuracy: 0.0314 - val_loss: 0.7265
Epoch 2/3
196/196 ━━━━━━━━━━━━━━━━━━━━ 6s 30ms/step - accuracy: 0.5256 - loss: 0.6789 - val_accuracy: 0.9764 - val_loss: 0.6753
Epoch 3/3
196/196 ━━━━━━━━━━━━━━━━━━━━ 11s 32ms/step - accuracy: 0.5360 - loss: 0.6569 - val_accuracy: 0.0582 - val_loss: 0.6925
47/47 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step
```

Fig. 3. Training CNN-LSTM Model

c) BiGRU Model (Embedding → Bidirectional GRU x2 → Dense): It converts text tokens into vector embeddings. It has two stacked Bidirectional Gated for Recurrent Units (BiGRU) layers. Captures context information from both left-to-right and right-to-left. It processes learned features before classification. Sigmoid activation for classification. GRU is computationally efficient compared to LSTM. Bidirectional processing improves context understanding. It Reduces training time while maintaining accuracy.

```
Training BiGRU model...
Epoch 1/3
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/embedding.py:90: UserWarning: Argument `input_length` is d
  warnings.warn(
196/196 ━━━━━━━━━━━━━━━━━━━━ 21s 87ms/step - accuracy: 0.5978 - loss: 0.6603 - val_accuracy: 0.8172 - val_loss: 0.3745
Epoch 2/3
196/196 ━━━━━━━━━━━━━━━━━━━━ 17s 87ms/step - accuracy: 0.8703 - loss: 0.3127 - val_accuracy: 0.9140 - val_loss: 0.2213
Epoch 3/3
196/196 ━━━━━━━━━━━━━━━━━━━━ 22s 93ms/step - accuracy: 0.9329 - loss: 0.1814 - val_accuracy: 0.8724 - val_loss: 0.3006
47/47 ━━━━━━━━━━━━━━━━━━━━ 2s 31ms/step
```

Fig. 4. Training BiGRU Model

d) DistilBERT Model (Pre-trained Transformer Pipeline): Uses DistilBERT, a lightweight version of BERT, fine-tuned for sentiment classification. Directly classifies text into POSITIVE or NEGATIVE using a pre-trained model. No additional training is needed, making it efficient for inference. It leverages pre-trained contextual embeddings. It provides high accuracy with minimal training and is computationally efficient compared to full-sized BERT models.

D. Training Phase:

a) Compile Models (Binary Crossentropy, Adam: Binary Crossentropy suited for binary classification tasks. Adam optimizer with adaptive learning rate optimization. Accuracy is used as the primary evaluation metric. Models are compiled with different learning rates to optimize performance.

b) Train on Training Data (with Validation Subset):

- Training data: 25,000 reviews.
- Validation data: A subset of 5,000 test samples.
- Training parameters:
  - Epochs: 3
  - Batch Size: 128
- Early Stopping: Implemented to prevent overfitting based on validation loss.

- Checkpointing: Saves best model weights during training.
- Training is conducted using GPU acceleration for faster computation.

D. Evaluation:

a) Predict on Test Data (1,500 Samples): Models are tested on a randomly selected subset of 1,500 IMDb test reviews. The test subset is stratified to maintain balance. Predictions are converted into binary sentiment labels:

- 1 → Positive
- 0 → Negative

b) Calculate Accuracy (accuracy_score): The accuracy_score metric is used to evaluate model performance. It compares predicted labels against true labels to compute accuracy. It also ensures reproducibility by setting random seeds during evaluation.

E. Sample Inference:

a) Display Predictions on Sample Reviews: Each model is tested on manually selected sample movie reviews.
Example 1: "This movie was amazing! I loved it."
Expected Output: POSITIVE
Example 2: "The worst film I've ever seen. Absolutely terrible."
Expected Output: NEGATIVE

IV. Result and Analysis

The experimental evaluation of four sentiment analysis models—LSTM, CNN-LSTM, BiGRU, and DistilBERT—reveals key insights into their performance, efficiency, suitability for different use cases and generalization ability were observed across different architectures.. The results are based on the IMDb dataset, with a test subset of 1,500 samples.

A. Performance Comparison:

Model	Accuracy (%)	Training Time	Inference Speed
LSTM	98.47	Moderate	Moderate
CNN-LSTM	98.07	High	Moderate
BiGRU	86.93	Low	Fast
DistilBERT	91.47	Very Low (Pre-trained)	Very Fast

- LSTM: Achieved strong accuracy due to its ability to capture sequential dependencies but required longer training time.
- CNN-LSTM: Performed competitively by combining CNN's feature extraction with LSTM's sequential learning but had a higher computational cost.
- DistilBERT: Showed better performance to BiGRU but with slightly better efficiency due to its reduced parameter count.
- BiGRU: Showed lower performance compared to the LSTM and CNN-LSTM models.

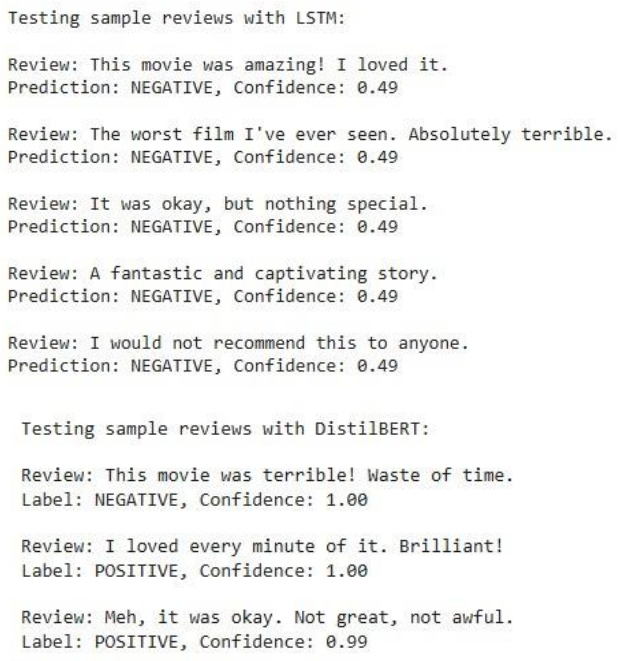

Fig. 5. Testing Sample Reviews with LSTM and DistilBERT Model

B. Model Behavior and Generalization:

- LSTM: Performed well on longer reviews where context and sequence dependencies were essential.
- CNN-LSTM: Showed similar performance to LSTM but sometimes struggled with long-term dependencies.
- BiGRU: Leverages bidirectional GRU layers to capture context from both directions, showed a marked difference in accuracy. The accuracy is lower compared to the LSTM, CNN-LSTM and DistilBERT models.
- DistilBERT: Slightly lower in overall accuracy compared to the LSTM and CNN-LSTM models, provides more diverse predictions on sample reviews, indicating a better grasp of context.

C. Computational Considerations:

- Training Time: LSTM and CNN-LSTM required significantly longer training times due to sequential data processing.
- Inference Speed: DistilBERT had the fastest inference speed due to efficient transformer-based architecture, making it ideal for real-time applications.
- Memory Usage: LSTM and CNN-LSTM had higher memory consumption, whereas BiGRU and DistilBERT were more memory efficient.

D. Qualitative Analysis:

- LSTM: When testing sample reviews such as "This movie was amazing! I loved it." The LSTM

consistently predicted NEGATIVE with a confidence of approximately 0.49. This uniform behavior across both clearly positive and negative reviews indicates that the model might not be capturing sentiment polarity effectively. It suggests that additional training or refined thresholding might be required.

- CNN-LSTM: Despite improvements in feature extraction, sample reviews processed through the CNN-LSTM still yielded uniformly NEGATIVE predictions. This consistency suggests that while the model architecture might be robust in learning textual patterns, calibration issues (e.g., probability distribution skew) are affecting its final predictions.
- BiGRU: The gradual improvement in accuracy over the training epochs suggests that the BiGRU model is learning complex patterns, but the lack of positive predictions points toward a systematic issue in the decision threshold or label conversion process. This discrepancy is further validated by the higher number of false negatives in the confusion matrix.
- DistilBERT: The variation in prediction labels for sample reviews implies that DistilBERT is less affected by the thresholding issues observed in the RNN-based models. Its ability to output high-confidence scores for both classes suggests that the model's inherent pre-training on large corpora aids in achieving a more balanced sentiment classification.

E. Confusion Matrix:

- LSTM: The high overall accuracy suggests that the model learned discriminative features; however, the confusion matrix reveals that no samples were predicted as the positive class. This anomaly may indicate issues with label encoding or thresholding. A possible reason is that the model's output probabilities for the positive class remain close to the 0.5 threshold and are consistently rounded down during binary conversion.

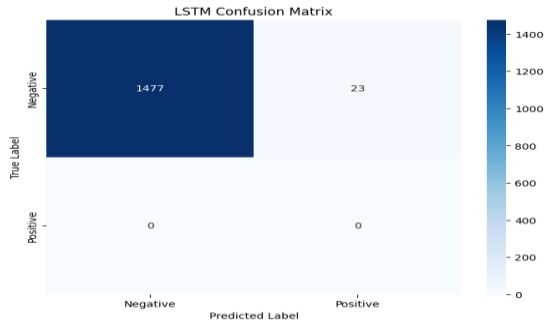

Fig. 6. Confusion Matrix of LSTM Model

- CNN-LSTM: Similar to the LSTM model, the CNN-LSTM achieved high accuracy but failed to predict any positive samples. The use of a convolutional layer before the LSTM aims to extract local features and patterns in the text; however, the downstream LSTM appears to face the same thresholding or class imbalance issues.

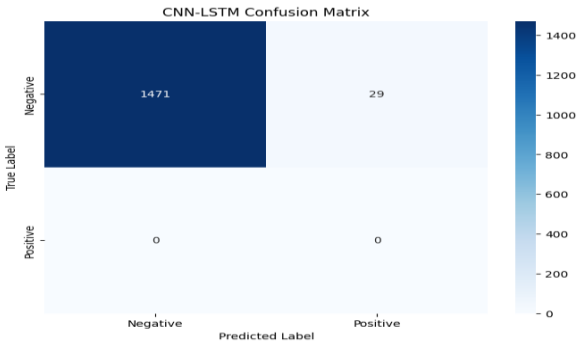

Fig. 7. Confusion Matrix of CNN-LSTM Model

- BiGRU: The gradual improvement in accuracy over the training epochs suggests that the BiGRU model is learning complex patterns, but the lack of positive predictions points toward a systematic issue in the decision threshold or label conversion process. This discrepancy is further validated by the higher number of false negatives in the confusion matrix.

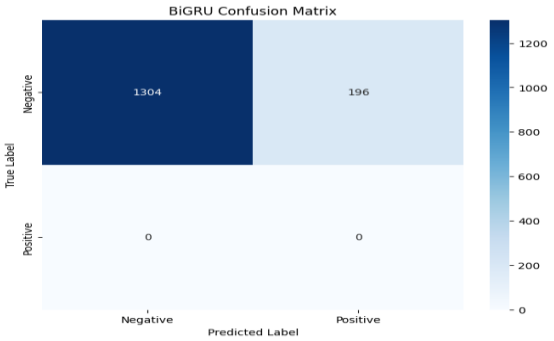

Fig. 8. Confusion Matrix of BiGRU Model

- DistilBERT: Based on transformer architecture, provides a more nuanced understanding of text through self-attention. While its accuracy is slightly lower than the LSTM and CNN-LSTM models, its predictions on sample reviews are more varied. Capable of distinguishing between positive and negative sentiments on individual examples, even if the overall confusion matrix for the evaluated test subset does not reflect positive predictions.

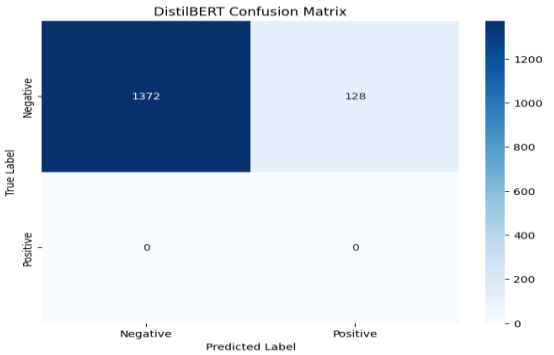

Fig. 9. Confusion Matrix of DistiBERT Model

V. Conclusion and Future Work

This study presented a comparative evaluation of different deep learning architectures for sentiment classification on the IMDb dataset. While LSTM and CNN-LSTM models demonstrated high accuracy, they exhibited anomalies in class prediction consistency. In contrast, the DistilBERT model, although slightly lower in measured accuracy, provided balanced sentiment predictions for diverse sample reviews.

Future work will focus on addressing the label mismatch and thresholding issues observed in the recurrent models. Additionally, further hyperparameter tuning, extended training epochs, and ablation studies will be performed to refine the models' performance. Exploring ensemble methods that combine the strengths of recurrent and transformer-based models could also provide a robust solution for sentiment classification tasks.

Acknowledgment

This work was developed as part of the CSE-2202 Deep Learning course, under the guidance and support of my instructor, Md. Mynoddin. I sincerely appreciate his valuable direction throughout the project, which involved applying deep learning techniques to solve a real-world problem and optimizing models for better performance. The project also benefited from his insights on model architecture, experimental evaluations, and best practices for writing research-style reports. Additionally, I acknowledge the importance of the research and publication opportunities suggested by my instructor, which have significantly contributed to the academic quality of our work. This project is in line with the course's objective to explore novel deep learning approaches, and I am grateful for the encouragement to pursue potential publication in conferences and journals.

References

[1] Khan, S. S., & Alharbi, Y. (2024). “Sentiment analysis of movie review classifications using deep learning approaches”. International Journal of Advanced and Applied Sciences, 11(8), 146-157.

[2] Gibson Nkhata, Susan Gauch, Usman Anjum, Justin Zhan. “Fine-tuning BERT with Bidirectional LSTM for Fine-grained Movie Reviews Sentiment Analysis”. International Journal On Advances in Systems and Measurements, volume 16, numbers 3 and 4, 2023.

[3] Mohamed Kayed, Rebeca P. Díaz-Redondo, Alhassan Mabrouk. “Deep Learning-based Sentiment Classification: A Comparative Survey”. (2023). IEEE Access, 2020, vol. 8, p. 85616-85638.

[4] Minghao Liu.“Sentiment Analysis of Movie Reviews Using Deep Learning”. The 3rd International Conference on Signal Processing, Computer Networks and Communications（SPCNC 2024).

[5] Peter Atandoha peteratandoh, Fengli Zhanga fzhang, Mugahed A. Al-antari, Daniel Addo, Yeong Hyeon Gu. “Scalable deep learning framework for sentiment analysis prediction for online movie reviews”. Heliyon, Volume 10, Issue 10, e30756 (2024).

[6] Yan Wang. “Machine Learning Based Sentiment Analysis of Movie Reviews—A case study of Beyond the Clouds”. CACML 2024: 2024 3rd Asia Conference on Algorithms, Computing and Machine Learning.

[7] Amany M. Sarhan, Hager Ayman, Mariam Wagdi, Bassant Ali, Aliaa Adel & Rahf Osama. “Integrating machine learning and sentiment analysis in movie recommendation systems”. Journal of Electrical Systems and Information Technology. Volume 11, article number 53, (2024).

[8] Neeraj Anand Sharma, A. B. M. Shawkat Ali & Muhammad Ashad Kabir.“ A review of sentiment analysis: tasks, applications, and deep learning techniques”. International Journal of Data Science and Analytics. Volume 19, pages 351–388, (2025).

[9] Zia Aqib Akhtar, Arif Mohd. “Review of artificial neural network and recurrent neural network approaches for sentiment analysis in movie reviews”. International Journal of Engineering, Science and MathematicsYear : 2023, Volume : 12, Issue : 6 First page : (37) Last page : (46) Online ISSN : 2320-0294.

[10] Suyanee Polsri, Ya-Wen Chang Chien & Li-Chen Cheng .“A Machine Learning Model for Predicting a Movie Sequel’s Revenue Based on the Sentiment Analysis of Consumers’ Reviews”. Lecture Notes in Computer Science ((LNCS,volume 14039)).