Journal on
Artificial Intelligence

Tech Science Press

DOI: 10.32604/jai.2023.045617

ARTICLE

Opinion Mining on Movie Reviews Based on Deep Learning Models

Mian Muhammad Danyal[1], **Muhammad Haseeb**[1], **Sarwar Shah Khan**[2,*], **Bilal Khan**[1] **and Subhan Ullah**[1]

[1]Department of Computer Science, City University of Science & Technology, Peshawar, 25000, Pakistan

[2]Department of Computer Science, Iqra National University, Swat, 19200, Pakistan

*Corresponding Author: Sarwar Shah Khan. Email: sskhan0092@gmail.com

Received: 02 September 2023 Accepted: 15 December 2023 Published: 31 January 2024

ABSTRACT

Movies reviews provide valuable insights that can help people decide which movies are worth watching and avoid wasting their time on movies they will not enjoy. Movie reviews may contain spoilers or reveal significant plot details, which can reduce the enjoyment of the movie for those who have not watched it yet. Additionally, the abundance of reviews may make it difficult for people to read them all at once, classifying all of the movie reviews will help in making this decision without wasting time reading them all. Opinion mining, also called sentiment analysis, is the process of identifying and extracting subjective information from textual data. This study introduces a sentiment analysis approach using advanced deep learning models: Extra-Long Neural Network (XLNet), Long Short-Term Memory (LSTM), and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM). XLNet understands the context of a word from both sides, which is helpful for capturing complex language patterns. LSTM performs better in modeling long-term dependencies, while CNN-LSTM combines local and global context for robust feature extraction. Deep learning models take advantage of their ability to extract complex linguistic patterns and contextual information from raw text data. We carefully cleaned the IMDb movie reviews dataset with the goal of optimizing the results of models used in the experiment. This involves eliminating unnecessary punctuation, links, hashtags, stop words, and duplicate reviews. Lemmatization is also used for keeping consistent word forms. This cleaned IMDb dataset is evaluated on the proposed model for sentiment analysis in which XLNet performs well achieving an impressive 93.74% accuracy on the IMDb Dataset. The findings highlight the effectiveness of deep learning models in improving sentiment analysis, showing its potential for wider applications in natural language processing.

KEYWORDS

Opinion mining; deep learning; XLNet; movie reviews; IMDb dataset

1 Introduction

Movies are an engaging form of entertainment that can both joys, teach, and educate us. They can help us understand different traditions and perspectives. They can help us in communicating with others and sharing our experiences. The movie industry expands year after year as a result of freshly produced and previously made movies. This is mostly because technology has improved rapidly, making it easier for people to discover and view movies from different places [1]. People may write and read movie reviews on the internet. Many websites currently provide information on movie reviews, including the Internet Movie Database (IMDB), Rotten Tomatoes, and others [2]. Movie reviews are the opinions or thoughts of people who enjoy the film. Combining all of these reviews helps people decide whether or not to watch the movie [3]. Movie reviews can be lengthy and detailed, making it challenging to read them all in a short period of time. This is especially true while choosing which movie to watch tonight [4]. Movie reviews may contain spoilers, ruining the experience for those who have not watched it. As a result, sentiment analysis can be used to classify movie reviews into positive and negative classes.

Sentiment analysis (SA), also known as opinion mining or sentiment mining, is a natural-language-processing (NLP) technique that can be used to extract subjective information from movie reviews [5]. It is used to classify movie reviews by analyzing the textual content and determining the overall opinion expressed, whether favorable or unfavorable. This can be achieved by identifying phrases and words related to positive or negative sentiment and then using algorithms for machine learning (ML) to classify the reviews correspondingly. Opinion extraction helps organizations such as the entertainment industry in gaining important insights into audience preferences, improve marketing techniques, make informed decisions, and increase overall viewer experience [6].

Deep learning uses a multi-layer method to the neural network's hidden layers. Conventional machine learning algorithms define and extract features either manually or through feature selection methods. Deep learning models, on the other hand, train and extract characteristics automatically, resulting in improved accuracy and performance. Deep learning models have been proven to be useful for a range of NLP tasks in recent years, including sentiment analysis, text summarization, and question-answering. Deep learning can understand complex connections between words and phrases, allowing it to overcome the challenges of natural language [7].

The paper introduces a sentiment analysis system that categorizes movie reviews as positive or negative. The system's performance is evaluated by comparing deep learning models XLNet, LSTM, and CNN-LSTM for the task of sentiment analysis. Our key focus is on quantifiable improvements over existing state-of-the-art models, with an emphasis on the enhanced accuracy achieved through the proposed methodology. We cleaned up the dataset and chose the hyperparameters settings carefully to get better results. The dataset used is the IMDB Dataset of 50 K reviews, a refined version of the original IMDB Dataset that has undergone several preprocessing steps to enhance its quality. These steps include data cleaning, lemmatization, and case normalization. During data cleaning, links, punctuation, hashtags, duplicate reviews, and stop words were removed. XLNet is a generalized autoregressive pretraining approach that learns bidirectional contexts by maximizing predicted probability across all factorization order permutations [8]. Long short-term memory (LSTM) is a recurrent neural network (RNN) architecture that excels at learning long-term dependencies in sequential data [9]. CNN-LSTM is a sequence modeling deep learning model that combines the strengths of convolutional neural networks (CNNs) with long short-term memory (LSTM) networks [10]. We contributed the following to this experiment:

- The dataset was pre-processed in a number of processes, including data cleansing, spelling correction and lemmatization, to improve results.
- The experiment analyses the results of implementing the proposed method for deep learning models, XLNet, LSTM, and CNN-LSTM on the IMDb Dataset over 5 epochs, with its main focus on evaluating each model's achieved accuracy in the discriminative and analytical classification of IMDb movie reviews.

- The hyperparameters, which include the maximum sequence length, learning rate, batch size, and number of epochs, are optimized to improve model performance.

The article is arranged as follows: The related work is discussed in Section 2. Section 3 contains the research methodology, while Section 5 has the results and discussion. Section 6 discusses the study's conclusion.

2 Related Work

Sentiment analysis plays an important role in guiding people's decisions regarding movie selection, helping them determine whether a particular movie is worth investing their time in. This technique involves extracting knowledge and evidence from text data, particularly positive and negative comments expressed in movie reviews. The primary objective of sentiment mining is to classify movie reviews as either positive, negative, or neutral based on the conveyed sentiment. This field has many significant research advancements in recent years, with the emergence of diverse methodological approaches. Abimanyu et al. [2] proposed a sentiment mining method for Rotten Tomatoes movie reviews. Logistic Regression (LR) was used for classification, Information Gain for feature selection, and TF-IDF for feature extraction. To begin, the approach pre-processes the movie reviews by removing stop words and stemming the words. Then it depends on Information Gain to choose features. They also discussed the reduction in accuracy caused by stemming. Their study provides valuable insights into the strengths and weaknesses of different approaches to sentiment mining.

BERT-CNN-based strategy was proposed by Zhang et al. to improve the performance of the original BERT model for sentiment analysis on the IMDb dataset in the context of movies and television. The purpose of analyzing audience reviews was to improve decision-making in fields such as casting and plot development. The BERT-CNN method was introduced expressly to tackle the complex challenge of sentiment mining for movie reviews [11]. Sharma et al. [12] developed a sentiment analysis of movie reviews using over 25,000 reviews from various sources. They implemented a machine learning model using Naive Bayes, Logistic Regression, and Support Vector Machine classifiers to classify movie reviews as good, negative, or neutral. They also deleted the stop words, and used stemming to minimize dimensionality. The study analyzed the three algorithms without discussing the results and provided suggestions on movie review classification.

Prayoga et al. [13] proposed a sentiment analysis framework for Indonesian movie reviews using the K-nearest neighbors (KNN) algorithm and chi-square feature selection. The experiments focused at the process used to choose the best K value, beginning with pre-processing, and how the K value of the KNN algorithm affected model performance. They observed that these factors had a significant impact on model performance. They recommended increasing the dataset, balancing the label distribution, researching KNN parameters, hyperparameter adjustment, and contrasting KNN with other classification techniques. Ali et al. [14] proposed an ensemble model for sentiment extraction of movie reviews using SVM and BERT. By combining the advantages of both models rather than doing it separately, they aimed to make predictions that were more accurate. When compared to the individual performance of the SVM and BERT models, the experimental findings showed better prediction accuracy, validating the ensemble model's effectiveness.

Dahir et al. [15] presented sentiment analysis for movie reviews using tokenization, lemmatization, and feature extraction algorithms such as Word of Bags and TF-IDF. They created categorization of sentiment models implementing Logistic Regression, SVM, and Random Forest. Sentiment mining accuracy and text analytics technologies were preferred. The research improves sentiment analysis and text analytics accuracy. Bodapati et al. [16] presents a sentiment analysis model for movie reviews that uses Long Short-Term Memory (LSTM) networks to address issues that traditional recurrent neural networks (RNNs) face. The focus is on implementing the categorization of reviews as positive or negative. The study monitors the effect of key hyperparameters such as dropout, layer count, and activation functions on model performance. The authors present results on different neural network configurations using the IMDb benchmark dataset, providing insights into the best configurations for effective sentiment analysis.

Rani et al. [17] proposed using convolutional neural networks (CNN) to analyze sentiment in Hindi movie reviews. They collected a dataset of movie reviews that had been carefully rated by native Hindi speakers and performed experiments with various CNN setups, changing the number and size of filters in the convolutional layers. The main challenge they faced was accurately classifying sentiment in Hindi movie reviews using deep learning techniques, which they successfully handled with their CNN-based methodology. Using natural language processing (NLP) technology. Zhu [18] developed a novel movie rating prediction algorithm. They used data mining techniques on a dataset of roughly 45,000 movies from Movielens that includes information such as movie duration, title, budget, story summary, genre, and more. The problem they faced was transforming a significant volume of textual input into vector data using NLP techniques so that machine learning algorithms like random forest and neural networks could be implemented. They aimed to achieve a fairly reliable model for forecasting movie ratings through significant tuning and experimenting. Naznin et al. [19] proposed novel strategies for improving unsupervised sentiment analysis in review comments. Their approach addressed many significant issues, including dealing with negations in text effectively, improving the placements of terms within sentences, and using the effects of exclamatory markings on sentiment. In the following step of their research, they used a cluster ensemble approach to analyze sentiment orientation. When compared to other strategies, this procedure resulted in a significant improvement in accuracy. The summary of overall literature review is discussed in Table 1.

Table 1: Summary of literature review

Reference and year	Techniques used	Preprocessing techniques used	Types of datasets used	Evaluation measures	Advantages and disadvantages of techniques
2023 [2]	Logistic regression method and information gain feature selection	Data cleaning, stemming, lemmatization, tokenization	Rotten Tomatoes movies reviews	Precision, recall, F1-score	Shows performance reduction due to stemming and lemmatization performing better
2023 [11]	BERT-CNN, logistic regression, BERT	Data cleaning	IMDb movie reviews	Accuracy, precision, recall, F1-score	BERT-CNN approach is better at negative sentiment detection

(Continued)

Table 1 (continued)

Reference and year	Techniques used	Preprocessing techniques used	Types of datasets used	Evaluation measures	Advantages and disadvantages of techniques
2023 [12]	Naive Bayes, logistic regression, and support vector machine	Data cleaning, stemming	IMDb dataset for movie reviews	Accuracy, precision, recall, F1-score	Quantitative evaluation with performance metrics
2023 [13]	TF-IDF, chi-square and KNN	Data cleaning, stemming, case folding	Indonesian-language movie review	Precision, recall, F1-score	Efficient feature selection
2023 [14]	LSTM, CNN-LSTM, MLP, CNN, SVM, NB	Data splitting	IMDb movie reviews	Accuracy	Improved results, use of hybrid model but computational complexity
2023 [15]	Logistic regression, random forest, SVM	Data cleaning, case folding, tokenization, lemmatization	IMDb movie reviews	Accuracy, precision, recall, AUC	Comprehensive approach and better accuracy
2022 [16]	LSTM-DNN, MLP, logistic regression	Data cleaning, tokenization	IMDb movie reviews	Accuracy	Long term dependency handling, effective sequence modelling
2018 [17]	CNN, KNN, ME, NB	Data cleaning	Hindi movie reviews	Precision recall F-measure kappa MAE RMSE	Improved accuracy, comparative analysis
2023 [18]	Random forest, BP network	Data cleaning	Movielens dataset	Accuracy, precision, recall, F1-score.	Improved accuracy, integration of advance models, resource intensive
2023 [19]	SVM, LR, DT, MNB, RF	Data cleaning, spelling correction	B. Pang movie reviews dataset	Accuracy	Handling negations, positional weighting, improved accuracy

The literature review explored diverse opinion mining techniques using models like Logistic Regression, BERT, Naive Bayes, SVM, TF-IDF, Chi-Square, KNN, LSTM, CNN, Random Forest, etc., evaluating with metrics such as Accuracy, Precision, Recall, F1-score, and AUC. Advantages and disadvantages were identified, including the impact of preprocessing methods. The proposed method on the IMDb dataset included thorough data cleaning (removing duplicates, stop words, punctuation, spelling correction, lemmatization, label encoding). Fine-tuned XLNet, LSTM, and CNN-LSTM models aimed at improving results, while the overall method focused on simplicity for reduced training time.

3 Methodology

The proposed method is divided into six stages: data collection, data preprocessing, train testing split, model hyperparameter optimization, XLNet, LSTM, CNN-LSTM implementation, and performance evaluation. The first stage involves collecting data from 50,000 IMDB reviews. In the second phase, the dataset is processed and cleaned in preparation for the models. The third phase divides training and testing data into 75% for training and 25% for testing. Hyperparameters are optimized in the fourth stage to improve model performance. Fig. 1 shows how XLNet, LSTM, and CNN-LSTM are used for sentiment mining in the fifth phase. Finally, in the sixth phase, the performance of the models is evaluated using several kinds of metrics to determine their effectiveness.

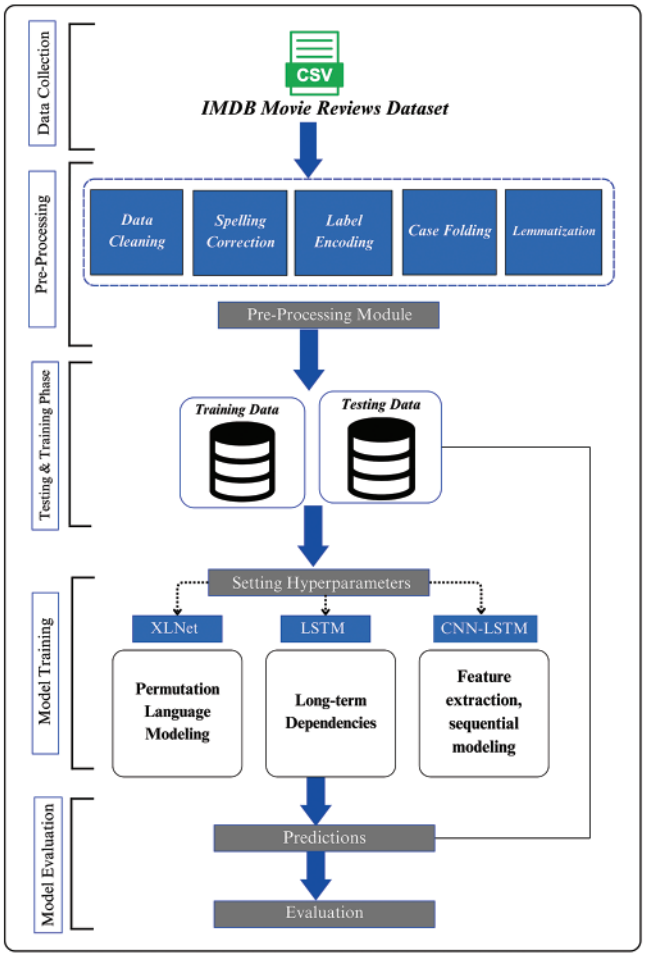


Figure 1: Systematic flow of the proposed method

Algorithm 1: Proposed Method

```
input: imdb_raw_movie_reviews_dataset
// Data Preprocessing
Step1. load_imdb_movie_reviews_dataset ()
Step2. preprocess_imdb_dataset ()
# Train-Test Split
Step3. split_data_into_training_and_testing_sets ()
# Parameter Tuning
Step4. Setting_hyperparameters ()
# Model Initialization
Step5. initialize_ model ()
# Model Training
Step6. train _model (training_dataset, model)
// Model Testing
Step7. evaluate_model (testing_dataset, model)
Step8. record_model_accuracy (model, model_accuracy_on_test_dataset)
// Results Analysis
Step9. identify_best_performing_model (model_accuracy_on_test_dataset)
output: trained_models, model_accuracy_on_test_dataset, Score {Final sentiment Score}; Sentiment {positive, negative}
```

3.1 Preprocessing

Pre-processing is the process of preparing a dataset for analysis after data collection [20]. Our pre-processing techniques include data cleaning, which removes punctuation, HTML tags, links, hashtags, and 432 duplicate reviews from the dataset. We also eliminated stop words, which are common words that add little meaning to the text, and correct incorrect spelling in the dataset using the text blob library [21].

After cleaning the data, lemmatization is performed instead of stemming because it provides better results [2]. Lemmatization is the process of putting together different forms of the same word that have changed over time and giving back the base form, also called the lemma [22]. Label encoding is a method to change the positive and negative sentiment in the dataset to numbers. It is used in natural language processing (NLP) to turn categorical data into number values so that it can be used by machine learning methods [23]. The negative reviews are labeled as 0 and positive reviews are labeled as 1.

To standardize the words used in reviews and make it simpler to process, case normalization changes all text to the same case, which is commonly lowercase [24]. Fig. 2 shows the word cloud created from the post-and pre-processed IMDB dataset. A word cloud is a graphic representation of text in which words are magnified in accordance with how frequently they occur. It gives a brief summary of the most common words.

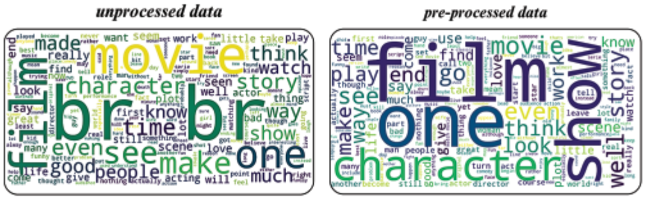


Figure 2: Word cloud

3.2 *Testing and Training*

In machine learning, it is important to split the data into two sets: a training set and a test set. The training set is used to train the model, while the test set is used to evaluate the model's performance [25]. In this experiment, 25% of the data was allocated for testing, while the remaining 75% of the data was used for training. This technique provided that the model's performance and accuracy could be tested without consideration for a specific class. In Fig. 3, the distribution of the testing data and training data is visualized.

25%
Test
75%
Train

Figure 3: Test train split

3.3 *Techniques*

In this paper, deep learning models such as XLNet, LSTM, and CNN-LSTM were used to perform sentiment mining of movie reviews. Deep learning models have significant advantages such as automatic feature learning, handling complex data, increased performance, and dealing with non-linear and structured or unstructured data [7]. XLNet is a powerful model with transformer architecture that performs well at long-range dependencies and noisy data management. LSTMs are suitable for sequential tasks, learning long-term dependencies, and dealing with variable-length input. CNN-LSTMs combine CNNs' local feature extraction and LSTMs' long-term learning for better natural-language-processing applications. These models significantly advance various fields, driven by their powerful capabilities and architectural designs. The following details provide more information.

3.3.1 XLNet

XLNet is an advanced method that combines the advantages of Autoregressive and Autoencoding methods through a technique called permutation language modeling. The neural architecture of XLNet is specifically designed to perform better at Autoregressive tasks, such as Transformer-XL and the carefully crafted two-stream attention mechanism. As a result, XLNet have better results from its previous pretraining methods on various tasks, showing superior performance [8].

The XLNet architecture for movie reviews classification in Fig. 4 takes a movie review text as input. The input text is passed to a transformer encoder, which produces a sequence of hidden states. These hidden states are then passed to a permutation language modeling module, which generates all permutations of the input text and predicts the masked tokens. The output of the permutation language modeling module is then passed to a classification layer, which predicts the sentiment of the movie review. The predicted sentiment of the movie review is the output of the XLNet architecture.

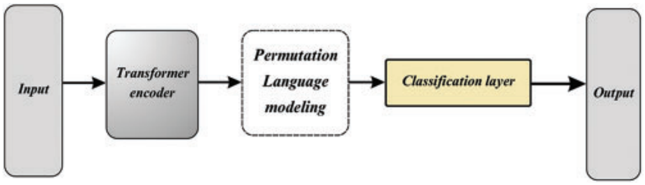


Figure 4: XLNet architecture

3.3.2 Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) is a recurrent neural network (RNN) architecture that improves tasks that require sequential data, such as movie review classification. Long-term dependencies between words in a review can be learned by LSTMs, which is important for accurately categorizing the sentiment of the review [9]. The reviews would be tokenized and transformed into a sequence of word IDs in a conventional LSTM-based movie reviews classification system. After that, the word IDs would be used to train an LSTM model. Based on the sequence of word IDs, the LSTM model would learn to predict the emotion of a review. The LSTM algorithm has transformed machine learning and neurocomputing. This is attributed to greatly improving Google's speech recognition capabilities, as well as the performance of machine translations in Google Translate [26].

Fig. 5 shows an LSTM model that began with an Embedding layer that converts inputs into dense vectors. Following that is a Dropout layer, which prevents overfitting by reducing specific neurons during training. Then, an LSTM layer learns the data's long-term dependencies. Following that, a Spatial Dropout1D layer drops whole 1D feature maps to provide common dropouts across time steps. Another LSTM layer refines the learning of sequential data, and the final prediction is made by the Output layer, resulting in a six-layer model.

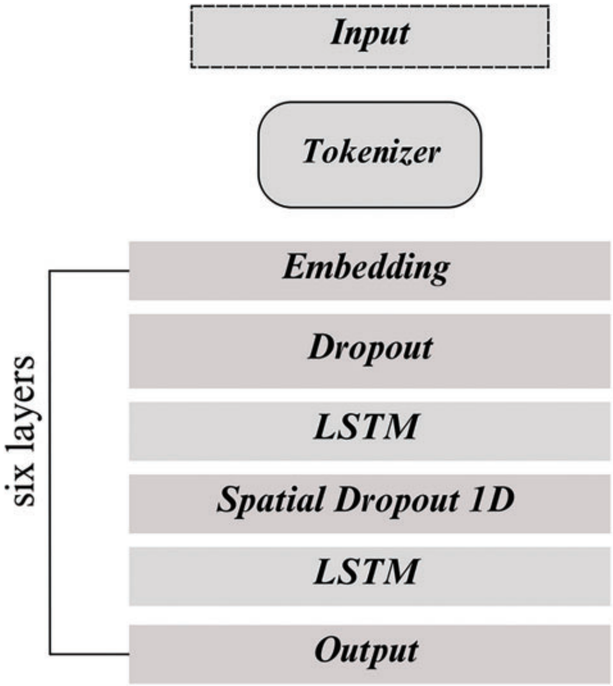


Figure 5: LSTM six layers architecture

3.3.3 CNN-LSTM

CNN-LSTM is a deep learning model that combines the strengths of convolutional neural networks and long short-term memory networks. CNNs are effective at extracting spatial information from data, but LSTMs are effective at learning long-term dependencies. CNN-LSTMs have been highlighted to perform well in a number of tasks, including NLP speech recognition and image classification [27].

Fig. 6 shows a CNN-LSTM hybrid model with an input layer that defines the form of the input data. An embedding layer consequently, which turns the input integer sequences into dense vector representations. The convolutional layer that follows uses filters to capture local patterns, while the max pooling layer decreases dimensionality. The dropout layer avoids overfitting while the LSTM layer captures sequential dependencies. Finally, using a sigmoid activation function, the output layer provides the predicted sentiment of the input sequence. This model combines the feature extraction powers of CNNs with the sequential understanding strengths of LSTMs, making it suitable for sentiment analysis.

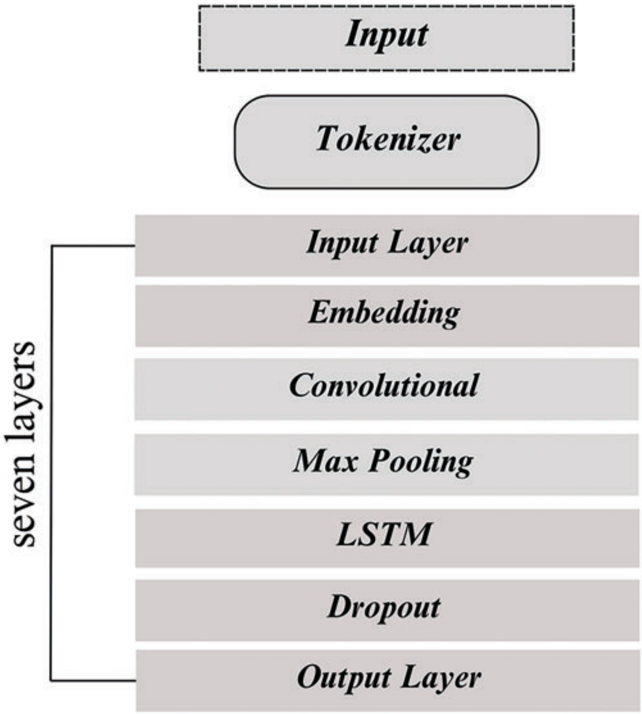


Figure 6: CNN-LSTM seven layers architecture

3.4 Hyperparameters Tuning

Hyper-parameters are parameters that are not learned from data during training and must be manually set before training begins. They represent higher-level configuration options for the ML algorithm, and their values are usually determined by the characteristics of the data being used and the algorithm's ability to learn from it. These hyper-parameters influence the learning algorithm's behavior and performance by acting as instructions or constraints [28].

Appropriate hyper-parameters have significance because they have a significant effect on the model's ability to generalize and make accurate predictions. The process of carefully experimenting with various combinations of hyper-parameter values in order to find the optimal settings that maximize the model's performance [29]. The hyperparameters tuned for this experiment are shown in Tables 2–4.

Table 2: Parameters of XLNet

Parameters	Value (XLNet)
Number of epoch	5
Maximum sequence length	128
Batch size	128
Learning rate	0.0001
Weight decay	0.01

Table 3: Parameters for CNN-LSTM

Parameters	Value
Number of epoch	5
Embedding layer (output dimensions)	128
Dropout	0.5
Batch size	128
Optimizer	Adam
MaxPooling1D (pool_size)	2

Table 4: Parameters for LSTM

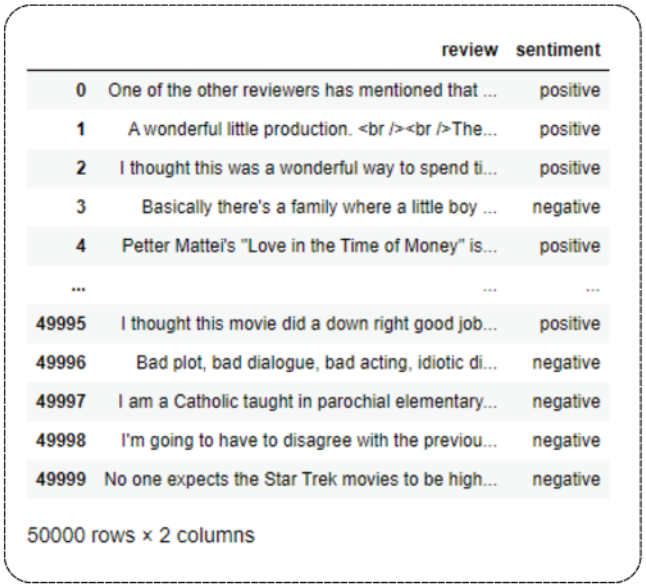
Parameters	Value
Maximum sequence length	128
Batch size	128
Embedding dimensions	128
LSTM units	128
LSTM dropout	0.2
LSTM recurrent_dropout	0.2
Optimizer	Adam
Number of epochs	5

4 Experimentation Setup

This study used Google Colab, a free cloud-based platform that allows users to run Python code in a web browser. The NVIDIA Tesla K80 GPU with 12 GB of RAM powers Google Colab. This GPU is better for many deep-learning tasks.

4.1 Dataset Description

The IMDb dataset of 50,000 reviews is a complete collection of user-written movie reviews. This dataset, which is used commonly in natural language processing and sentiment analysis studies, has a wide range of opinions on different movies. Each review contains sentiments such as positive or negative, which gives more background for figuring out how people feel about a movie [30]. The snapshot of the dataset used is shown in Fig. 7.



	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
...	...	...
49995	I thought this movie did a down right good job...	positive
49996	Bad plot, bad dialogue, bad acting, idiotic di...	negative
49997	I am a Catholic taught in parochial elementary...	negative
49998	I'm going to have to disagree with the previou...	negative
49999	No one expects the Star Trek movies to be high...	negative

50000 rows × 2 columns

Figure 7: IMDB dataset snapshot

The distribution of classes in the IMDb dataset is shown in Fig. 8.

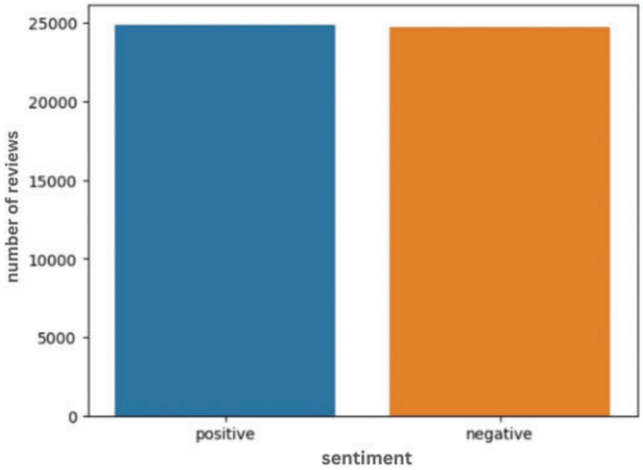


Figure 8: Distribution of classes in IMDB dataset

4.2 Performance Metrics

In this research experiment, the precision, recall, F1-score, accuracy, and evaluation time (in seconds) are used to measure performance. The effectiveness of a system model is evaluated using a confusion matrix. In confusion matrix, TP is true positive (positive predicted and factual), FN is false negative (negative predicted and factual), FP is false positive (positive predicted and negative factual), and TN is true negative (negative predicted and factual). Here are more specifics:

4.2.1 Accuracy

The accuracy of the predictions is determined by dividing the number of predicted reviews by the total number of reviews [31].

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

4.2.2 Precision

Precision is determined by dividing the number of reviews that were correctly predicted as positive by the total number of reviews that were predicted to be positive.

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

4.2.3 Recall

Recall is measured by dividing the number of correctly predicted positive reviews by the total number of positive reviews.

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

4.2.4 F1-Score

The F1-measure, also known as the f-score or f-measure, calculates the performance of an algorithm by taking both precision and recall into account. The F1-measure equation is shown below:

$$\text{F1} - \text{Score} = \frac{2 * Precision * recall}{Precision + recall} \tag{4}$$

4.2.5 Training Time

The training time measures the duration for all steps in each epoch across each model.

5 Result and Discussion

This section contains the experiment's results and analysis of the models used. Table 5 shows the performance of XLNet over 5 epochs, Table 6 shows the performance of LSTM, and Table 7 shows the performance of CNN-LSTM models on the IMDB dataset. The models with bold results performed well in their respective measures in each table.

Table 5: Performance of XLNet using IMDB dataset

Model	No. of epochs	Precision	Recall	F1-score	Accuracy	Loss	Training time
XLNet	1	86.72%	92.63%	89.58%	89.11%	36.94%	843 s
	2	87.49%	93.67%	90.48%	90.04%	32.44%	866 s
	3	89.14%	93.21%	91.13%	90.83%	23.93%	**831 s**
	4	88.72%	93.97%	91.27%	90.92%	20.20%	849 s
	5	**91.48%**	**96.54%**	**93.94%**	**93.74%**	**20.15%**	850 s

Table 6: Performance of LSTM using IMDB dataset

Model	No. of epochs	Precision	Recall	F1-score	Accuracy	Loss	Training time
LSTM	1	87.94%	83.39%	85.61%	85.91%	**33.19%**	321 s
	2	85.85%	**89.82%**	87.79%	87.44%	37.80%	294 s
	3	90.61%	80.78%	85.41%	86.14%	50.52%	265 s
	4	**92.86%**	79.17%	85.47%	86.47%	41.35%	**236 s**
	5	86.77%	89.41%	**88.07%**	**87.83%**	37.60%	240 s

Table 7: Performance of CNN-LSTM using IMDB dataset

Model	No. of epochs	Precision	Recall	F1-score	Accuracy	Loss	Training time
CNN-LSTM	1	90.63%	86%	88.23%	88.48%	38.89%	538 s
	2	**92.01%**	80.94%	86.11%	86.89%	14.17%	528 s
	3	85.81%	**89.64%**	87.68%	87.35%	6.53%	510 s
	4	86.07%	89.01%	87.52%	87.24%	**2.41%**	508 s
	5	90.71%	86.75%	**88.68%**	**88.76%**	3.04%	514 s

Table 5 shows the performance of XLNet on the IMDB dataset across different training epochs. The model's precision, recall, F1-score, and accuracy all improve as the number of epochs grows. This improvement is predicted when the model is exposed to more data and its representations are improved. The improvement in performance from 1 to 5 epochs shows that XLNet benefits from more extended training and can capture complex structures and patterns in the IMDB dataset, resulting in better classification results. The training time remains relatively stable across all of the epochs, indicating that longer training durations could lead to further improvements in performance, but the trade-off between training time and performance must be considered. The performance of XLNet in bar chart is shown in Fig. 9.

XLNet performed well on the IMDB dataset as it is a self-supervised learning model that combines masked language modelling and relative position encoding. XLNet was able to learn the underlying statistical features of the language, the connections between words and phrases, and the arrangement of words in a sentence. The performance of LSTM model on IMDB Dataset is shown in Table 6.

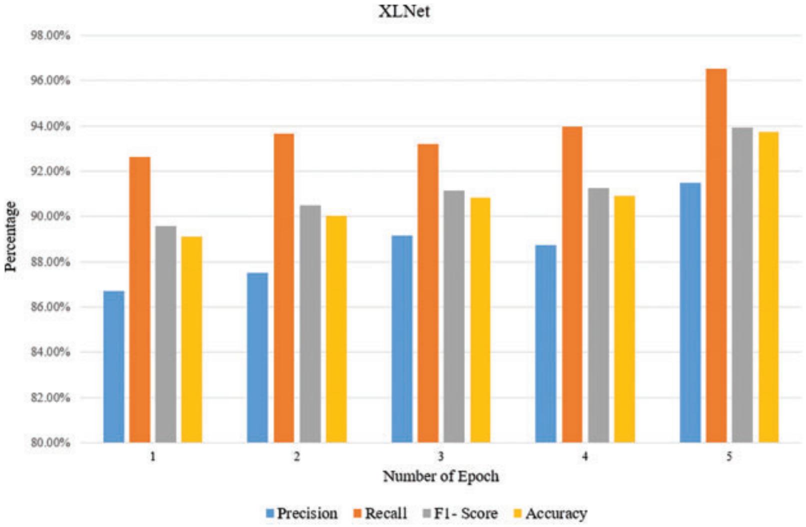


Figure 9: Performance of XLNet on IMDB dataset using bar chart

Table 6 shows the performance of an LSTM model on the IMDb dataset across five separate epochs. As the number of epochs increases, the model's performance improves in certain measures while fluctuating in others. The model achieves appropriately good accuracy and recall, showing that it can produce accurate positive predictions and capture a considerable number of real positive data. The F1-score, on the other hand, shows that there may be a balance between accuracy and recall since it fluctuates over epochs. The loss measure indicates the model's performance on the training data, and it usually lowers as the epochs go, showing that the model is learning. The training time lowers with each epoch, indicating quicker training as the model increases performance. The performance of LSTM in bar chart is shown in Fig. 10.

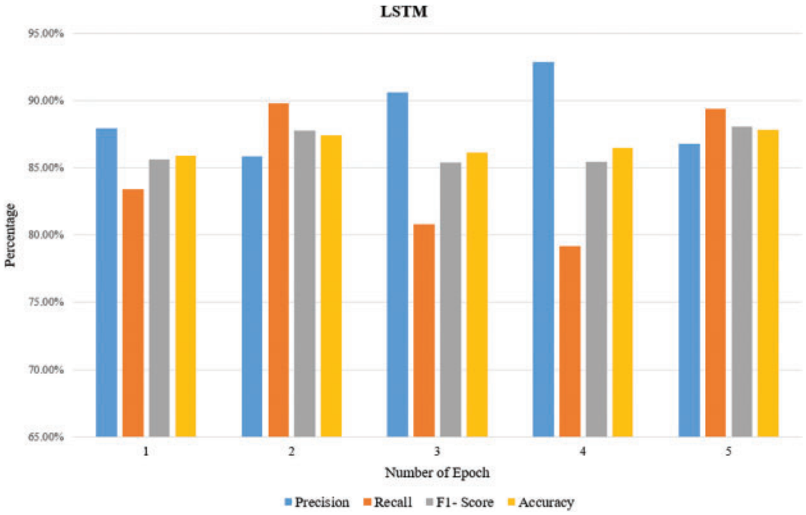


Figure 10: Performance of LSTM on IMDB dataset using bar chart

LSTM performs reasonably well on the IMDb dataset with accuracy of 87.83%, but there could be some overfitting issues as the training time decreases with more epochs, showing that the model starts memorizing the data rather than generalizing effectively. Due to XLNet's larger training dataset, its performance is 5% better than LSTM's. The results of CNN-LSTM on IMDB Dataset are shown in Table 7.

Table 7 shows the performance of the CNN-LSTM model on the IMDB dataset over five different epochs. CNN-LSTM achieves relatively high precision, recall, F1-score, and accuracy across all epochs, showing that it's capable to classify movie reviews as positive or negative sentiment.

In the first epoch, the model achieves 90.63% precision, 86% recall, and 88.23% F1-score, with an accuracy of 88.48% and a loss of 38.89% in 538 s of training time. The model's precision increases slightly in the following epochs, and highest at 92.01% in epoch two, but its recall fluctuates, impacting the F1-score. The performance of LSTM in bar chart is shown in Fig. 11.

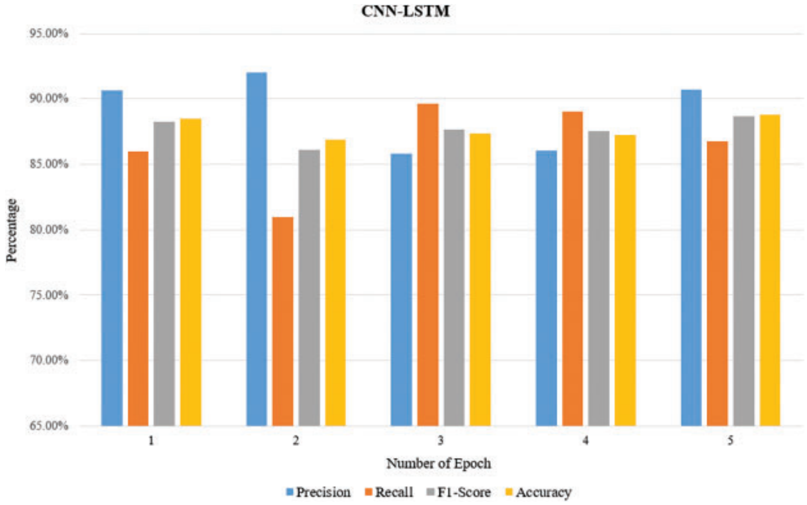


Figure 11: Performance of CNN-LSTM on IMDB dataset using bar chart

The CNN-LSTM's performance shows that it effectively captures both spatial and temporal features from the input data, making it a suitable choice for text classification tasks. It outperforms LSTM by 1% in terms of accuracy but 5% less compared to the more advanced XLNet model. As the model learns, the training time decreases gradually. The training time ranges from 508 to 538 s, which is considered efficient for this model's performance on the IMDB dataset.

5.1 Discussion

The performance of XLNet was better with accuracy of 93.74% on the proposed method which is 5% higher as compared to LSTM and CNN-LSTM as a result of its advanced transformer architecture, bidirectional context understanding, pre-training on a large dataset, transfer learning, and improved attention mechanism. These features enable XLNet to capture long-term dependencies in information, get a more robust understanding of linguistic patterns, and focus on key information while disregarding noise, resulting in improved sentiment analysis results on the IMDB dataset. Fig. 12 shows a bar chart comparing the accuracy of models using the proposed method.

XLNet required more time to train, averaging 848 s. LSTM and CNN-LSTM models, on the other hand, trained faster, with average times of roughly 284 and 514 s, respectively. Although XLNet performed better in many ways, its longer training time may be an issue in circumstances when we need faster results but have limited resources or time to train the model.

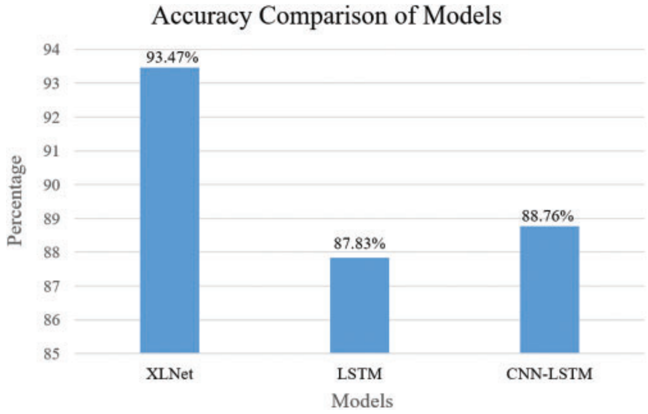


Figure 12: Accuracy comparison of models

We limited the number of epochs to 5 for all models because the models' performance did not improve significantly after this point. There can be a number of possible explanations for these results, including overfitting, dataset characteristics, and "the models having already learned the underlying patterns during their first epochs".

Our model is designed to be both efficient and performant. However, to reduce training time, more computational power is required. This will allow to train the model more effectively and quickly. Additionally, it is worth noting that the sensitivity of the hyperparameters in our proposed method could impact its performance. Fine-tuning these settings becomes important, as small adjustments may impact the overall effectiveness of the model. The performance comparison of proposed method with existing techniques are shown in Table 8.

Table 8: Performance comparison of existing techniques with proposed method

S. no.	Models	Methods	Accuracy
1	Zhang et al. [11]	BERT-CNN	92.9%
2	Dahir et al. [15]	SVM	89.20%
3	Bodapati et al. [16]	LSTM + DNN	88.46%
4	Sahu et al. [32]	Random Forest	88.95%
5	Ali et al. [14]	CNN-LSTM	89.20%
6	**Proposed**	**XLNet**	**93.74%**

5.2 Limitations

This study used XLNet and LSTM, and CNN-LSTM models to conduct sentiment analysis on IMDb movie reviews. When considering the results of this study, the limitations should be considered. The data used may not fully represent all types of movie reviews, and data bias is possible. Furthermore, due to resource constraints, the reliance on pre-trained models such as XLNet may make it less accessible to some researchers. The study focused mainly on English-language movie reviews, and the models may not perform similarly on other languages due to linguistic and cultural differences. Furthermore, XLNet training requires more computational power and time. Finally, while the reported results are helpful, differences may occur under different conditions. Despite these limitations, the study provides helpful insights into sentiment analysis of movie reviews; however, these limitations must be acknowledged for a complete understanding.

6 Conclusion and Future Work

Movie reviews allow people to participate in discussions and share their thoughts and feelings about the movie with other people who enjoyed it, creating a feeling of community and connection. Our study has provided valuable insights into the field of opinion mining on movie reviews. We have highlighted the capabilities of deep learning models such as XLNet, LSTM, and CNN-LSTM in effectively classifying and predicting movie reviewers' sentiments. The maximum accuracy achieved by XLNet, 93.74%, shows its effectiveness as a natural language processing strategy. Furthermore, while LSTM and CNN-LSTM did not achieve the same level of accuracy as XLNet, they still produced useful sentiment classification results. The results of this study show that sentiment analysis is an effective technique for classifying movie reviews. There are some limitations to this study that must be acknowledged. Our analysis focused primarily on English-language movie reviews, and the models may not perform as effectively when applied to other languages due to linguistic variations and cultural differences. Additionally, XLNet training requires more computational power and time. When implementing XLNet, researchers with limited computational resources may face challenges. Our future work in the field of sentiment analysis on movie reviews will include studying and improving the performance of models for analysing sentiments in movie reviews in languages other than English, with the goal of increasing the applicability and accuracy of sentiment extraction in multilingual contexts.

Acknowledgement: We would like to express our sincere gratitude to all the individuals and organizations who have contributed to the successful completion of this research paper. We also acknowledge the contribution of our participants, who provided valuable feedback and suggestions that helped to refine this research methodology and enhance the quality of the findings. We would like to thank the participants who generously gave their time and participated in our study, without whom this research would not have been possible. Thank you all for your contributions to this research paper.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: SSK, MH and MMD; data collection: MMD, MH; analysis and interpretation of results: MMD, SSK, SU and BK; draft manuscript preparation: MMD, MH, BK, SU and SSK. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. R. Nugraha, M. D. Purbolaksono, and W. Astuti, "Sentiment analysis on movie review from rotten tomatoes using modified balanced random forest method and Word2Vec," *Build. Inf., Technol. Sci.*, vol. 5, pp. 153–161, 2023.

[2] A. J. Abimanyu, M. Dwifebri, and W. Astuti, "Sentiment analysis on movie review from rotten tomatoes using logistic regression and information gain feature selection," *Build. Inf., Technol. Sci.*, vol. 5, pp. 162–170, 2023.

[3] A. Rashad, M. Khan, Y. Ghadi, H. Aljuaid, and Z. Nawaz, "A deep neural network-based approach for sentiment analysis of movie reviews," *Complex.*, vol. 2022, pp. 5217491, 2022. doi: 10.1155/2022/5217491.

[4] M. M. Danyal, S. S. Khan, M. Khan, M. B. Ghaffar, B. Khan, and M. Arshad, "Sentiment analysis based on linear support vector machine performance and multinomial Naïve Bayes using movie reviews with baseline techniques," *J. Big Data*, vol. 5, pp. 1–18, 2023.

[5] S. S. Khan, M. Khan, Q. Ran, and R. Naseem, "Challenges in opinion mining, comprehensive review," *A Sci. Technol. J.*, vol. 33, pp. 123–135, 2018.

[6] M. Khan, M. S. Khan, and Y. Alharbi, "Text mining challenges and applications—A comprehensive review," *Int. J. Comput. Net. Inf. Secur.*, vol. 20, no. 12, pp. 138, 2020.

[7] N. C. Dang, M. N. Moreno-García, and F. de la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electron.*, vol. 9, no. 3, pp. 483, 2020.

[8] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdino, and Q. V. Le, "XLNet Generalized autoregressive pretraining for language understanding," arXiv preprint arXiv:1906.08237, 2019.

[9] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D: Nonlinear Phenom.*, vol. 404, pp. 132306, 2020.

[10] C. I. Garcia, F. Grasso, A. Luchetta, M. C. Piccirilli, L. Paolucci, and G. Talluri, "A comparison of power quality disturbance detection and classification methods using CNN, LSTM and CNN-LSTM," *Appl. Sci.*, vol. 10, no. 19, pp. 6755, 2020.

[11] B. Zhang, "A BERT-CNN based approach on movie review sentiment analysis," in *2023 8th Int. Conf. on Soc. Sci. Eco. Develop.*, vol. 163, 2023. doi: 10.1051/shsconf/202316304007.

[12] H. Sharma, S. Pangaonkar, R. Gunjan, and P. Rokade, "Sentimental analysis of movie reviews using machine learning," in *2nd Int. Conf. Data Sci. Intell. Appl. (ICDSIA-2023)*, vol. 53, 2023. doi: 10.1051/itm-conf/20235302006.

[13] I. Prayoga, M. Deifebri, and P. Adiwijaya, "Sentiment analysis on indonesian movie review using KNN method with the implementation of chi-square feature selection," *J. Media Inform. Budidarma*, vol. 7, no. 1, pp. 369–375, 2023.

[14] N. M. Ali, M. M. A. E. Hamid, and A. Youssif, "Sentiment analysis for movies reviews dataset using deep learning models," *Int. J. Data Min. Knowl. Manag. Process*, vol. 9, pp. 19–27, 2019.

[15] U. M. Dahir and K. A. Faisal, "Utilizing machine learning for sentiment analysis of IMDB movie review data," *Int. J. Eng. Trends Technol.*, vol. 71, pp. 18–26, 2023.

[16] J. D. Bodapati, N. Veeranjaneyulu, and S. N. Shareef, "Sentiment analysis from movie reviews using LSTMs," *Ingénierie des Syst. d'Inf.*, vol. 24, no. 1, pp. 125–129, 2019.

[17] S. Rani and P. Kumar, "Deep learning based sentiment analysis using convolution neural network," *Arab. J. Sci. Eng.*, vol. 44, pp. 3305–3314, 2019.

[18] H. Zhu, "Sentiment analysis of movies based on natural language processing," in *2023 4th Int. Conf. on Educ., Knowl. Inf. Manag. (ICEKIM 2023)*, Nanjing, China, Jun. 2023, pp. 1232–1240.

[19] F. Naznin and A. K. Mahanta, "Techniques for improving the performance of unsupervised approach to sentiment analysis," *Indones. J. Electr. Eng. Inform.*, vol. 11, no. 2, pp. 402–415, 2023.

[20] A. Alduailej and A. Alothaim, "AraXLNet: Pre-trained language model for sentiment analysis of Arabic," *J. Big Data*, vol. 9, pp. 72, 2022. doi: 10.1186/s40537-022-00625-z.

[21] S. Alam and N. Yao, "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis," *Comput. Math. Organ. Theory*, vol. 25, pp. 319–335, 2019.

[22] D. Khyani, B. S. Siddhartha, N. M. Niveditha, and B. M. Divya, "An interpretation of lemmatization and stemming in natural language processing," *J. Univ. Shanghai Sci. Technol.*, vol. 22, no. 10, pp. 350–357, 2021.

[23] V. Andersson, "Machine learning in logistics: Machine learning algorithms: Data pre-processing and machine learning algorithms," *J. Logist.*, pp. 23, 2017.

[24] A. S. Manek, P. D. Shenoy, M. C. Mohan, and K. R. Venugopal, “Aspect term extraction for sentiment analysis in large movie reviews using gini index feature selection method and SVM classifier,” *World Wide Web*, vol. 20, no. 2, pp. 135–154, 2017.

[25] J. Tan, J. Yang, S. Wu, G. Chen, and J. Zhao, “A critical look at the current train/test split in machine learning,” arXiv preprint arXiv:2106.04525, 2021.

[26] G. van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artif. Intell. Rev.*, vol. 53, pp. 5929–5955, 2020.

[27] R. Mutegeki and D. S. Han, “A CNN-LSTM approach to human activity recognition,” in *2020 Int. Conf. on Artif. Intell. Inf. Commun. (ICAIIC)*, Fukuoka, Japan, 2020, pp. 362–366.

[28] T. Agrawal, “Introduction to hyperparameters,” in *Hyperparameter Optimization in Machine Learning: Make Your Machine Learning and Deep Learning Models More Efficient*, New York, NY, USA: Apress, 2021, pp. 4–5.

[29] L. Yang and A. Shami, “On hyperparameter optimization of machine learning algorithms: Theory and practice,” *Neurocomput.*, vol. 415, pp. 295–316, 2020.

[30] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proc. of the 49th Annual Meet. Assoc. Comput. Linguist.: Hum. Lang. Technol.*, Portland, Oregon, USA, 2011, pp. 142–150.

[31] B. Khan, M. Arshad, and S. S. Khan, “Comparative analysis of machine learning models for PDF malware detection: Evaluating different training and testing criteria,” *J. Cybersecur.*, vol. 5, pp. 1–11, 2023.

[32] T. P. Sahu and S. Ahuja, “Sentiment analysis of movie reviews: A study on feature selection & classification algorithms,” in *2016 Int. Conf. on Microelectron., Comput. Commun. (MicroCom)*, Durgapur, India, 2016, pp. 1–6.