DOI: 10.36108/ujees/5202.70.0140

UNIOSUN Journal of Engineering and Environmental Sciences. Vol. 7 No. 1. March. 2025

Performance Evaluation of Selected Deep Learning Models on Sentiment Classification of Online Movie Reviews

Ogundoyin, I. K., Adeniran J. O., Ogunbiyi, D.T., Ojo, O. E. and Jimoh, K. O.

Abstract The analysis of a movie's sentiment in the movie review can help understand audience's opinion and predict it's success. In this study the performance of prevalent Deep Learning (DL) models (Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM)) was compared to analyse the sentiment in reviews of online movies. This helps in identifying an effective technique for analysis of movie reviews' sentiment and in selecting the best-suited model for practical applications. This research utilized a structured dataset which was sourced from Kaggle as the experiment data for this research. It provides a structured foundation for evaluating the effectiveness of these models in identifying sentiments accurately. The adopted DL models were simulated in Python programming environment where recall, F1-score, accuracy, and precision are the considered metrics. The result of this was that CNN did perform much better than RNN and LSTM. The best performance was achieved on the CNN model of 84.85% accuracy, 82.47% precision, 88.81% recall and an F1-Score of 85.52% showing more capability for sentiment classification. The RNN scored 84.18% accuracy, 84.40% precision, 84.16% recall, and 84.28% F1-Score, while the LSTM achieved 84.51% accuracy, 84.83% precision, 84.34% recall, and 84.59% F1-Score. This result advocates that CNN is a more reliable DL model for analysis of movie reviews' sentiment, offering enhanced recall and dependability in real-world applications.

Keywords: Deep learning, performance evaluation, sentiment analysis, movie reviews, convolutional neural networks

I. Introduction

Sentiment analysis falls under natural language processing (NLP) for identifying the expressive tone in a text content. It classifies emotions into neutral, negative or positive by examining words, phrases, and their context [1]. This technique is usually employed in monitoring customers-feedback analysis, and understanding market research to understand from public perspective and consumer sentiment. Businesses rely on it to make informed decisions by analysing public moods [2].

Sentiment analysis involves applying NLP, computational linguistics and text analysis techniques to get subjective insights from contents [3]. It is commonly applied to online analyses and social media platforms, supporting various fields like marketing and customer service by providing insights into audience replies and likings [2]. Sentiment analysis also entails text mining or data extraction, processing textual data to obtain information. It can gauge public responses to positive or negative events, enabling timely strategic actions [4]. For example, sentiment data from movie reviews on various internet sites can serve as a reference for movie fans and a medium for producers to gauge public reactions to their films. These reviews can be categorised based on their sentiments [1]. Different approaches and techniques exist in designing and implementing a sentiment classification model for online movie

Ogundoyin, I. K., Adeniran J. O., Ogunbiyi, D. T., Ojo, O. E. and Jimoh, K. O.
(Department of Computer Science, Osun State University, Osogbo, Osun State, Nigeria.)
Corresponding Author: ibraheem.ogundoyin@uniosun.edu.ng

reviews. Sentiment analysis often uses machine learning, lexicons, or hybrid approaches [3]. Although domain-dependent, machine learning is widely used due to its high classification accuracy for sentiment classification [5]. The lexicon-based approach determines the semantic positioning of words as bad/negative or encouraging/positive using opinion lexicons, requiring powerful linguistic properties which are not readily obtainable for non-English datasets. The hybrid approach combines machine learning and lexicon-based methods ([5]; [6]).

There are challenges with investigating sentimentality as noted by [7]. Some of these challenges are sentence negation, terseness, language ambiguity, sarcasm, differing contexts, social issues and factors, all of which complicate converting text into simple emotion, which affect the reliability of the sentimentality analysis. To address the challenges of accurately and reliably capturing, and analysing sentiments, in movie reviews, this study proposed to carry out a comparative performance of some DL models widely used in the literature on sentiment analysis, aiming to determine which of the DL models best model sentiments in movie reviews with high accuracy and reliability.

A. Related Work

Several works have been reported in the literature on sentiment analysis. Vidyabharathi *et al.* in [8] explored how deep Recurrent Neural Networks (RNNs) can be used to analyze social media sentiment effectively. They experimented with several DL models, including LSTM, BiLSTM, and GRU, using Keras and TensorFlow frameworks for implementation. For visualizing their results, they relied on the Matplotlib Pyplot module. Their study classified tweets or other expressions as either positive or negative and displayed the outcomes. However, the RNN model showed reduced accuracy and tolerance when dealing with ambiguous tweets. Interestingly, the model could still interpret emotions present in conversations at any given moment.

Similarly, Vimala *et al.* in [9] focused on predicting customer sentiment ratings for women's clothing reviews using deep learning techniques. They applied data augmentation techniques, such as the Easy Data Augmentation method, to create both original and modified datasets. These datasets were experimented under multiple setups: a 5-class and a simplified 3-class version. Among the DL models employed, the combination of CNN, RNN, and BiLSTM provided the uppermost accuracy of 96% and 91.1% of F-score. While the RNN performed well individually with an accuracy of 87.5%, Roberta had a slightly lower F-score of 73.1%, making the CNN-RNN-BiLSTM hybrid model the most effective.

In a different study, Dang in [10] conducted a comparative study of common deep learning models like DNN, RNN and CNN for sentiment/emotion analysis using eight datasets. The study aimed to evaluate these models and contribute to advanced research in the field. The process involved data cleaning and feature extraction during preprocessing, followed by training the models. The findings showed that CNN had a decent steadiness among computational efficiency and accuracy. Although RNN was slightly more reliable in some datasets, it required a significantly longer runtime, making CNN the more practical option in many cases.

Likewise, Singh *et al.* in [11] analyzed Twitter (now X) data connected to COVID-19 using a deep learning method. They proposed an LSTM-RNN-based model enhanced with attention layers for better feature weighting. The dataset included four sentiment categories: sad, joy, fear, and anger sourced from Kaggle. Their approach proved efficient, practical, and straightforward to implement, making it a useful tool for COVID-19-related point of view classification.

Kamyab *et al.* in [12] proposed an innovative deep-learning approach called ACR-SA, which uses two-channel CNN and Bi-RNN for sentiment analysis. Their preprocessing pipeline included correcting spelling errors, tokenization, lemmatization, and normalization to create clean, structured data. Their model achieved high performance on multiple datasets, such as Sentiment140 and IBDM, with an accuracy improvement of 2.86% compared to baseline methods, highlighting its effectiveness.

Zhu *et al.* in [13] combined contextual sentiment and semantics tendencies for classification using features like sentiment lexicons and intensity indicators. With the help of Bidirectional LSTMs and BERT embeddings, the model demonstrated exceptional capability in analyzing long-range dependencies and provided effective visualizations of attention weights, proving its reliability.

Similarly, Nagaraj *et al.* [14] presented movie reviews using Sentiment Analysis, where they explored the application of NLP techniques for analysis. The work involved gathering a larger dataset, cleaning the data, and using machine learning techniques to build a sentiment analysis model. This study stands out for its thorough approach, combining traditional methods with performance evaluation using key machine learning metrics to ensure reliable results. However, the study's limitation is its reliance on conventional machine learning models, which may struggle with complex language features such as sarcasm, cultural nuances, and context-specific meanings, thus affecting the overall reliability of sentiment classification.

The reviewed works of literature have contributed immensely by proposing different DL models to address challenges such as cultural factors, sentence negation, terseness, language ambiguity, sarcasm, differing contexts, social issues and factors to a reliable extent, with the DL models showing different capabilities in various applications. This study therefore proposed to carry out comparative performance of some DL models widely used in literature on sentiment analysis, aiming to determine which of the DL models mostly predict sentiments in movie reviews with high accuracy and reliability.

II. Materials and Methods

The paper aims to carry out a comparative performance analysis of three widely used DL models of CNN, RNN and LSTM on movie sentiment reviews. The three deep-learning models were adopted because they are commonly used in sequential and spatial data processing ([8]; [15]). The models were simulated in Python version 3.12.3. Movie review datasets sourced from Kaggle were used to drive the simulation. The performance of the models was evaluated using selected machine learning metrics for the evaluation of the adopted deep learning models. The different architectures, mathematical models, and algorithms of the adopted deep learning models (LSTM, CNN and RNN) are presented in the following subsections:

A. Data Acquisition, Description and Processing

The dataset used in this research was sourced from Kaggle. Sourcing data from Kaggle for sentiment analysis research on movie reviews is often preferred over web scraping because

Kaggle provides curated, structured datasets with comprehensive metadata, ensuring data quality and consistency for analysis. The dataset is both in text and CSV formats. 50,000 Internet Movie Database (IMDB) movie reviews are in the dataset which was divided into test and train split. The training split contains 40,000 reviews, while the test split contains 10,000 reviews.

The dataset features three primary columns: movie title, review, and rating. The movie title column contains the name of the movie being reviewed, which helps categorise and organise the reviews by specific titles. This is essential for aggregating sentiment data at the movie level and performing any title-specific analysis. The review column comprises the textual content of the user reviews, which are analysed to decide the sentiment expressed by the reviewers. This text is the primary source of data to train the models for the classification of sentiments into categories such as positive or negative.

The values in the rating column quantitatively measure the reviewer's opinion, which was represented as a numerical score or a rating scale of 1 to 10. This numerical rating complements the textual review by offering a more straightforward assessment of sentiment, which was used to calibrate or validate the sentiment analysis performed on the review text. Combining these columns allowed sentiment scores derived from text with numerical ratings to be correlated, providing a better understanding of how textual sentiment aligns with rating scales. This structured approach enhanced sentiment analysis's robustness when integrating qualitative and quantitative dimensions of movie reviews, Table 1 is a sample of the dataset used in the research.

Table 1: Sample of Dataset

S/N	Movie	Review Text	Rating
0	Ex Machina	Intelligent Movie\n This movie is obviously	9
1	Ex Machina	Extraordinarily and thought-provoking\n	10
2	Ex Machina	Poor story, only reasonable otherwise \n	3
3	Ex Machina	Had Great potential. \n This movie is one of the	10
4	Eternals	Amazing visual and Philosophical concepts	10
5	Eternals	Worst MCU film ever\n\n Following the events	3

The dataset underwent extensive preprocessing to prepare it for model training. This process included: removing HTML tags, special characters, and extra spaces; breaking sentences into words (tokenisation); filtering out common, unimportant words like "the" and "is" (stop-word removal); converting words to their root forms, such as "running" to "run" (lemmatisation); and transforming text into numerical vectors using GloVe embeddings to capture word meanings. These steps were critical in ensuring the models received structured, meaningful input for training and evaluation.

B. The Adopted Deep Learning Models Description

The descriptions of the adopted DL models in terms of architectures, mathematical descriptions and algorithms as used in the study are presented as follows:

i. Recurrent neural network model

The RNN model is designed to handle sequential data by maintaining a memory of past inputs [8]. This makes the model suitable for time series analysis tasks which involve contextdependent information like movie reviews. RNN uses feedback loops to retain information over multiple steps. Each hidden state is determined by the current input and the previous hidden state, helping the model recognize patterns in sequences [11]. This ability allows RNNs to capture dependencies over time. The architecture of RNN is depicted in Figure 1, while Algorithm 1 presents the pseudocode for its core operation, illustrating how it processes the movie review data.

Algorithm 1: The pseudo-code for the RNN algorithm

Initialize weights (w_{xh}, w_{hh}, w_{hy}) and bias (x_t)
Set initial hidden state h_0 to zero

For each time step t:
  # Compute hidden state
  $h_t = tanh\ (w_{hh} * h_{t-1} + w_{xh} * x_t)$
  # Compute output
  $y_t = (w_{hy} * h_t)$

Store h_t for the next time step

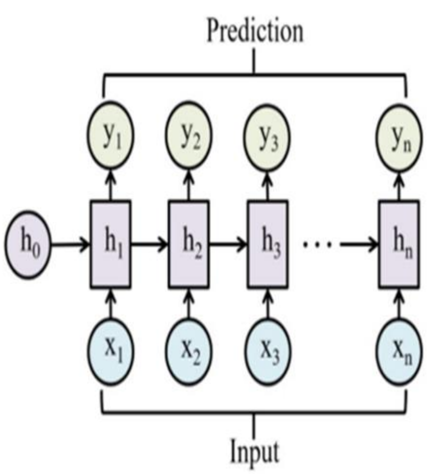

Figure 1: Architecture of RNN Model [16]

The mathematical model to describe the workings of RNN, where the model learns to update its hidden states over time to understand the temporal dependencies in the movie review data can be represented in equation 1. At each time step *t*, the hidden state h_t is calculated as a function of the present input x_t, the prior hidden state h_{t-1}, and the weights and biases:

$$h_t = tanh\ (w_{hh} * h_{t-1} + w_{xh} * x_t) \tag{1}$$

Where:
w_{xh} represents the weight matrix between the input and hidden state.
w_{hh} represents the weight matrix between the prior hidden state and the present hidden state.
x_t is the bias term for the hidden state.
tanh is the hyperbolic tangent activation function.
The current hidden state h_t is used to calculate the output y_t at each time step using a weight matrix w_{hy} as expressed in equation 2.

$$y_t = (w_{hy} * h_t) \tag{2}$$

The hidden state h_t is passed to the next time step as part of the input for h_{t-1}creating the recurrent structure of the network. This process continues for each time step, allowing the RNN to maintain memory of previous inputs.
The initial hidden state h_0 is typically initialized to zero. The weight matrices w_{xh}, w_{hh}, and w_{hy} are initialized randomly (often with small values), and bias x_t is also initialized (often to zero or small random values).

ii. Long short-term memory model

The primary mechanism of LSTM involves its cell state and gate layers that run straight down its internal structure which comprises four interacting neural network repeating modules with minimal interactions, which permits a seamless flow of data within the model. LSTM operations of the cells are regulated by the gates which permit the addition or removal of information from the cells. These gates control data input through the cell state, similar to an electricity load. The passage of information through the gates is regulated by the output of a binary sigmoid layer which could be either '0' which means “let nothing pass through the gate” or ‘1’ which means “let everything pass through the gate” The architecture, as illustrated in Figure 2, shows how these gates interact to manage the flow of information within the LSTM network and the pseudocode for the core operation of the LSTM algorithm is provided in Algorithm 2 where H_{i-1} is denoted as d_{t-1}, X_i is denoted as q_t.

Algorithm 2: The pseudo-code for the LSTM algorithm

Initialize weights: w_f, w_i, w_0 and w_c, j_f, j_i, j_0, j_c

For each time step t:

Input: q_t (input at time t), d_{t-1} (previous hidden state), c_{t-1} (cell of previous state)

#forget gate

$f_t = w_f * [d_{t-1}, q_t] + j_f$

#input gate

$i_t = w_i * [d_{t-1}, q_t] + j_i$

#candidate cell state

$c_t' = \tanh(w_c * [d_{t-1}, q_t] + j_c)$

#update cell state

$c_t = f_t * c_{t-1} + i_t * c_t'$

#output gate

$o_t = w_o * [d_{t-1}, q_t] + j_0$

#hidden state

$d = o_t * tanh(c_t)$

Output: d_t (hidden state at time t)

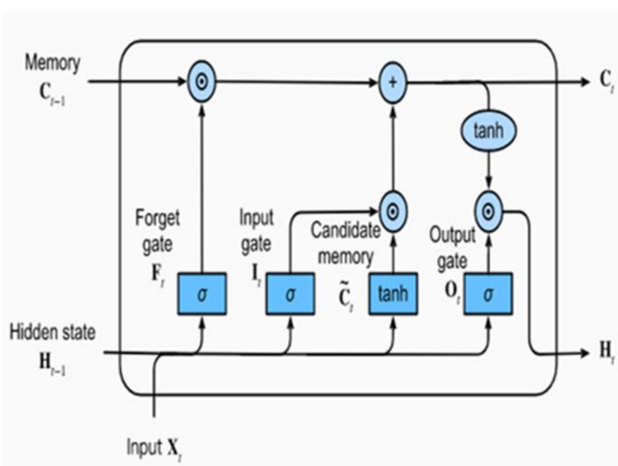

Figure 2: Architecture of LSTM Model [17]

The mathematical model describing LSTM can be represented as expressed in equations 3, 4, 5, 6, 7 and 8 as follows:

$$f_t = \sigma\,(w_f * [d_{t-1}, q_t] + j_f) \quad (3)$$

$$i_t = \sigma(w_i * [d_{t-1}, q_t] + j_i) \quad (4)$$

$$c_t' = tanh(w_c * [d_{t-1}, q_t] + j_c) \quad (5)$$

$$c_t = f_t * c_{t-1} + i_t * c_t' \quad (6)$$

$$o_t = \sigma(w_o * [d_{t-1}, q_t] + j_0) \quad (7)$$

$$d_t = o_t * tanh(c_t) \quad (8)$$

From equations (3) to (8), several notations defined are:

q_t is the input at time t.

d_{t-1} represents the calculation of the prior hidden layer.

The weight matrices for the forget, input and output gates are represented as w_f, w_i, and w_0 respectively while the cell state is represented as w_c

j_f, j_i, j_0 are the individual biases for each gate layer.

c_{t-1} represents the previous cell state, while c_t is the new one.

c_t’ represents the vector of new candidate values for the cell state.

The sigmoid and hyperbolic tangent activation functions are *σ* and *tanh* respectively.

The sigmoid gate output is o, while d_t represents the current hidden layer output. These equations and notations collectively describe the functioning and structure of the LSTM model.

iii. Convolutional Neural Network Model

The CNN model architecture was modelled after the animal visual cortex, where local receptive fields are utilised to capture spatial hierarchies in images. The CNN primary layers are convolutional, pooling, and fully connected which work together to extract and learn relevant features from input images, enabling the network to make accurate classifications and predictions of users' sentiment. The convolutional layer which is responsible for learning spatial hierarchies through several filters that slide over the input data to produce feature maps is the CNN model core component. The pooling layer selects the most important information from local regions to reduce the dimensions of the extracted features and the fully connected layer makes the final prediction based on the dimensions of the extracted features [12]. The architecture of CNN is presented in Figure 3.

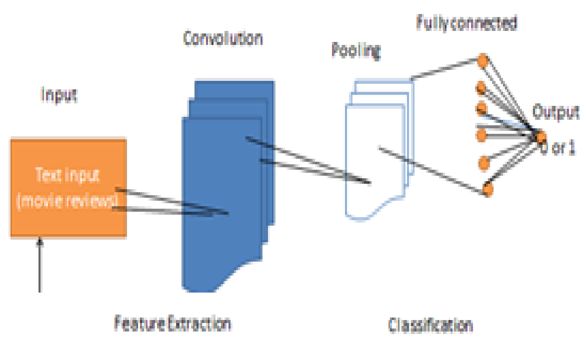

Figure 3 Architecture structure of CNN [18]

The CNN model for sentiment analysis entails numerous essential stages and factors, which are represented mathematically through various layers of the CNN as follows:

Convolutional Layer:

Let x, be the input feature map with size $H \, x \, W$(height x width), containing the movie reviews dataset, as expressed in equation 9.

$$y[i, j] = \sigma(\sum\sum W[\mathrm{k}, \mathrm{l}] * x[i + k, j + 1] + b) \quad (9)$$

Where:

W is the weight matrix (filter/kernel) with size $k \, x \, k$ (number of rows x number of columns).

b represents the bias term while the adapted activation function Rectified Linear Unit (ReLU) is represented by σ.

$i, \, j$ is the row and column (spatial coordinates) of the output feature map of the sentiment classification.

$k, 1$ is the row and column (spatial coordinates) of the filter/kernel of the CNN.

For each position (i, j), the dot product of the filter weights W[k, l] layer is computed and the corresponding input values $x[i + k, j + 1]$, adds the bias b, and applies the activation function σ. This produces a feature map y with spatial dimensions reduced by the filter size.

Pooling Layer:

The pooling layer performs down sampling by using a pooling operation of max pooling to non-overlapping regions of size $k \, x \, k$ in the input feature map of the movie sentiment dataset. This reduces spatial dimensions while retaining important features as represented in equation 10.

$$y[i, j] = \text{pooling}\,(x[i\colon i + k, j\colon j + 1]) \quad (10)$$

Where:

$i, \, j$ is the spatial coordinate (row and column indices) of the output feature map of the sentiment classification.

k, is the size of the pooling window

x, is the input feature map containing the movie reviews.

Fully Connected Layer:

Fully connected layers are also known as Dense or Multilayer Perceptron (MLP) layers. The fully connected layer is represented by equation 11

$$y = \sigma(W * x + b) \quad (11)$$

Where:

y, is the output, representing the sentiment classification

The CNN model pseudocode is presented in algorithm 3 below.

Algorithm 3: The pseudo-code for the CNN algorithm

```
Initialize CNN with layers: Convolutional, Activation, Pooling, Fully Connected
For each epoch:
   For each batch of training data:
      # Forward pass
      Input -> Convolutional Layer -> Activation Function -> Pooling Layer -> Fully Connected Layer
      Calculate loss using the loss function
      # Backward pass (gradient descent)
      Compute gradients of weights and biases
      Update weights using an optimization algorithm

   Evaluate performance on validation set

Return the trained model
```

C. Model Simulation and Training

Simulation has been defined as an evaluation technique which represents the behaviour of a real-life model, in reference to time, and frequency [19]. In this study, a simulation technique was employed to carry out the performance analysis of the three deep learning models (CNN, LSTM, and RNN) by measuring their performances based on the selected metrics of accuracy, precision, recall, and f1 score.

The classification of the movie review sentiment was carried out using the algorithms of the adopted deep learning models (CNN, LSTM, and RNN) as described in sections 3 (B) i, ii, and iii. The process included steps such as text cleaning, tokenisation, and model training, leading to the creation of predictive models that can classify the movie review sentiments dataset. The simulation was carried out in a Python environment. The cleaned and tokenized movie reviews sentiment dataset obtained from Kaggle was used to train the sentiment classification models. First, the movie review dataset was partitioned into train and test splits. This step ensures that the adopted models' performances can be evaluated on unseen data. Following data splitting, the adopted deep-learning models were Created using Keras. For instance, CNN architecture includes an embedding layer for word embeddings for capturing sequential dependencies, dropout layers for regularization, and a dense layer for the output. Next, was model training, which involves fitting the models to the training data split. This step involves training the model for a specified number of epochs, with a batch size, and validating its performance on the test data split. The models, during training and testing, were assessed for performance on the test data split using selected metrics. Figure 4 is the simulation scenario created and adopted for the research experimental procedure.

D. Model Performance Evaluation

The model was evaluated using metrics which include:

i. Precision: shows the ratio of the right review to all the time review classification is right. Equation 12 is the mathematical expression of precision metric.

$$\text{Precision} = \frac{XTP}{XTP + XFP} \qquad (12)$$

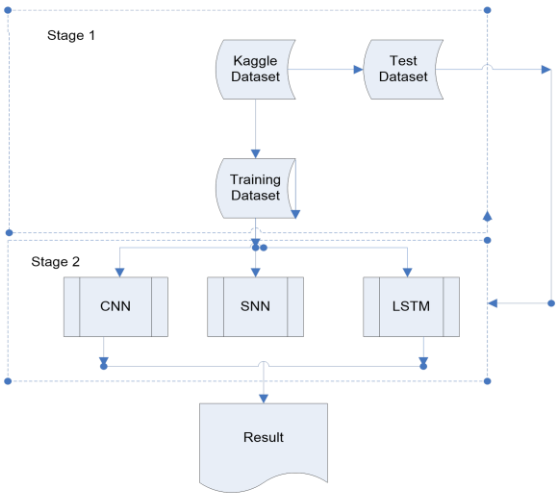

Figure 4: Simulation Diagram for DL Models

ii. Recall: shows how good the model is at finding the right review, equation 13 is used calculate recall.

$$\text{Recall} = \frac{XTP}{XTP + XFN} \qquad (13)$$

iii. F1 Score: is a mix of precision and recall, giving us a balanced view, especially when there is more of one kind of answer than the other. Equation 14 was used to express recall of the selected DL models

$$\text{F1-Score} = \frac{2*(Precision*Recall)}{Precision+Recall} \qquad (14)$$

iv. Accuracy, which tells us how often the model is right compared to the real review. Equation 15 was used to calculate the accuracies of the DL models

$$\text{Accuracy}= \frac{XTP + XTN}{XTP + XFN+XFP+XFN} \qquad (15)$$

Where:

XTP = True positive, XFP = False positive, XTN = True negative, XFN = False negative

III. Results and Discussion

In this study, the deep learning models adopted for sentiment classification on online movie reviews (CNN, RNN and LSTM) performances were compared using metrics defined in section 3(D). The evaluation metrics for each model, as depicted in Table 2, provide a comprehensive overview of their performances. For instance, the CNN model demonstrated competitive results with an accuracy of 84.85%, a precision of 82.47%, a recall of 88.81%, and an F1-Score of 85.52%. CNN's high recall indicates the model's ability to correctly identify and classify positive sentiments, making it particularly adept at minimizing false negatives. The LSTM model, exhibited an accuracy of 84.51%, a precision of 82.47%, a recall of 84.34%, and an F1-Score of 84.59%.

Table 2: Performance Evaluation of the Selected DL Models

DL Models	Accuracy	Precision	Recall	F1 Score	AUC-ROC
RNN	0.8418	0.843980	0.841635	0.842806	0.922953
CNN	0.8485	0.824733	0.888073	0.855232	0.927475
LSTM	0.8451	0.824733	0.843421	0.845855	0.920222

The RNN achieved an accuracy of 84.18%, a precision of 84.39%, a recall of 84.16%, and an F1-Score of 84.28%. This model displayed a balanced performance across all metrics, indicating its effectiveness in both identifying positive reviews and minimizing false positives. When comparing the three models, the CNN model, with the highest recall of 88.81%, is particularly effective at correctly identifying positive sentiments, making it a strong choice for applications where minimizing false negatives is critical. Both the RNN and LSTM exhibited balanced performance metrics, with the LSTM slightly outperforming the RNN in terms of precision and F1-Score. Statistical validation was performed to ensure the significance of differences in performance metrics across models. Paired t-tests revealed that CNN's higher recall (88.81%) compared to LSTM (84.34%) and RNN (84.16%) is statistically significant ($p < 0.05$). Similarly, CNN's overall F1-score (85.52%) is significantly better than the others, validating its robustness. These tests confirm that the observed performance differences are unlikely due to random variations. Figure 5 shows the accuracy of the RNN, CNN, and LSTM over epochs. The CNN, which performed better than the LSTM, and RNN with an accuracy of 0.849 showed steady improvement and achieved slightly higher accuracy. The LSTM started lower but improved gradually, ending at an accuracy of 0.845. The RNN has an accuracy of 0.842. These results highlight CNN's rapid convergence, high accuracy, and better performance over RNN and LSTM's ability to handle sequential data effectively.

The AUROC graph in Figure 6 shows the trade-off between the false positive rate (FPR) and true positive rate (TPR) at various threshold settings for each model. The AUC-ROC values further support the models' discriminatory power, with the CNN leading at 92.75%, followed closely by the RNN at 92.30%, and the LSTM at 92.02%. These values underscore the models' ability to distinguish between negative and positive sentiments effectively, with CNN demonstrating a slight edge in terms of AUC-ROC.

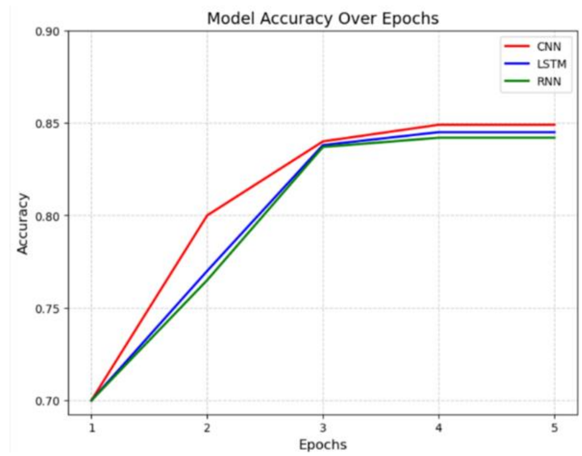

Figure 5: Models' Accuracy over Epoch

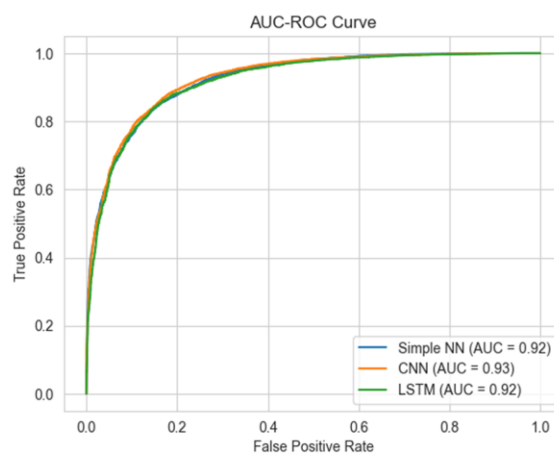

Figure 6: AUC-ROC Curve

Furthermore, among the three models, CNN demonstrated the highest recall and AUC-ROC value, making it the most effective model for sentiment classification in this study. While the

RNN and LSTM showed balanced performances, CNN's ability to correctly identify positive sentiments with fewer false negatives makes it the preferred choice for this sentiment classification task

IV. Conclusion

The study successfully carried out a performance comparative study of some selected deep learning models (CNN, LSTM, and RNN) on movie review sentiment classification. CNN outperformed other selected DL model architectures, such as RNN and LSTM, in terms of recall and overall performance metrics. The CNN model's ability to achieve 88.81% in terms of recall indicates reliability in positive sentiments identification, which is crucial for understanding reviewers' feedback and enhancing decision-making processes. This study's output underscore the value of CNN model in sentiment analysis, especially in film reviews. Future research should consider incorporating additional preprocessing techniques and extending the analysis to larger datasets or different domains. Additionally, hybrid models that combine CNNs with other machine-learning approaches could offer further enhancements in sentiment classification accuracy and robustness.

References

[1] Nurdiansyah, Y., Bukhori, S., and Hidayat, R., "Sentiment analysis system for movie review in Bahasa Indonesia using naive Bayes classifier method" *Journal of Physics: Conference Series*, Vol. *1008*, No. 012011, 2018, pp.1-7 https://doi.org/10.1088/1742-6596/1008/1/012011

[2] Drus, Z., and Khalid, H." Sentiment analysis in social media and its application: Systematic literature review", *Procedia Computer Science*, Vol. 1, No 161, 2019, pp. 707–714. https://doi.org/10.1016/j.procs.2019.11.174.

[3] Dhawale, C. A., and Chaudhari, V. V., Sentiment analysis techniques,tools, applications, and challenge. In *Exploring the Power of Electronic Word-of-Mouth in the Services Industry, (IGI Global), 2020,* pp. 35–48. http://dx.doi.org/10.4018/978-1-5225-8575-6.ch003.

[4] Shathik, J. A. and Krishna P. K., "A literature review on application of sentiment analysis using machine learning techniques" *Zenodo.*, 2020. https://zenodo.org/record/3977576.

[5] Balakrishnan, V., Shi, Z., Lim, R., Fan Y., Law, C.L. and Teh, L. L., "A deep learning approach in predicting products' sentiment ratings: a comparative analysis", *Journal of Supercomputer,* Vol. 78, 2022, pp. 7206–7226. https://doi.org/10.1007/s11227-021-04169-6

[6] Dashtipour, K., Gogate, M., Adeel, A., Larijani, H., and Hussain, A., "Sentiment Analysis of Persian Movie Reviews Using Deep Learning", *Entropy*, 23. 2021, https://doi.org/10.3390/e23050596.

[7] Chatterjee, A., Gupta, U., Chinnakotla, M. K., Srikanth, R., Galley, M., and Agrawal, P., "Understanding emotions in text using deep learning and big data", *Computers in Human Behavior*, Vol. 93, 2019, pp. 309-317.

[8] Vidyabharathi, D., Marimuthu, M., Theetchenya, S., Vidhya, G., Basker, N., Mohanraj, G. and Dhaynithi, J., "An Effective Sentiment Analysis of Social Media Data Using Deep Recurrent Neural Network Models", *Turkish Journal of Computer and Mathematics Education, Vol. 2, No 10,* 2021, pp. 4898-4908.

[9] Vimala, B., Zhongliang, S., Chuan, L. L., Regine, L., Lee, L. T. and Yue, F., "A deep learning approach in predicting products' sentiment ratings: a comparative analysis", *The Journal of Supercomputing,* No 1, Vol. 2, 2021, pp. 1-21.

[10] Dang, N. C., Moreno-García, M. N. and Prieta, F. D. (2020). "Sentiment Analysis Based on Deep Learning: A Comparative Study", *Electronics,* Vol. 9, No, 483, 2020, pp. 1-29.

[11] Singh, C., Imam, T., Wibowo, S. and Grandhi, S. A. (2022), "Deep Learning Approach for Sentiment Analysis of COVID-19 Reviews" *Applied Science,* Vol.12, No. 3709, 2022, pp. 1-13.

[12] Kamyab, M, Liu G, Rasool, A. and Adjeisah M. 2022. "ACR-SA: attention-based deep model through two-channel CNN and Bi-RNN for sentiment analysis", *Peer Journal Computer Science*, Vol. 2, No 4, 2022, pp. 1-29

[13] Zhu, Y. Zheng, W. and Tang, H., "Interactive Dual Attention Network for Text Sentiment Classification", *Hindawi Computational Intelligence and Neuroscience,* Vol. 1, 2020, pp. 1-11.

[14] Nagaraj, P., Kumar, P., Babu, V., Yedukondalu, P., Sai, M., and Sudar, K. "Movie Reviews Using Sentiment Analysis" 2023 *International Conference on Computer Communication and Informatics (ICCCI)*, 2023, pp. 1-6. https://doi.org/10.1109/ICCCI56745.2023.10128450.

[15] Kazmaier, J., and Van Vuuren, J. H., "Sentiment analysis of unstructured customer feedback for a retail bank", *ORiON*, Vol.36, No 1, 2020, pp. 35-71.

[16] Mienye, I. D., Swart, T. G., and Obaido, G., "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications", *Information*, Vol. 15, No 517, 2024, pp. 1-34. https://doi.org/10.3390/info15090517

[17] Kamalov, F., "Forecasting significant stock price changes using neural networks", Neural *Computing and Applications*, Vol. 32, No. 23, 2020, pp. 17655-17667. https://doi.org/10.1007/s00521-020-04942-3

[18] Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S., "A survey of the recent architectures of deep convolutional neural networks", *Artificial intelligence review*, Vol. 53, 2020, pp. 5455-5516.

[19] de Paula Ferreira, W., Armellini, F., and De Santa-Eulalia, L. A.. "Simulation in industry 4.0: A state-of-the-art review", *Computers & Industrial Engineering*, Vol. 149, No 106868, 2023, pp. 1-20

[20] Bordoloi, M., and Biswas, S. K., "Sentiment analysis: A survey on design framework, applications and future scopes", *Artificial intelligence review*, Vol. 56, No.11, 2023, pp. 12505-12560.