DESIGN AND SIMULATION OF ONLINE MOVIE REVIEWS SENTIMENT CLASSIFICATION MODEL

Ibrahim OGUNDOYIN[1], Dimple OGUNBIYI[2], John ADENIRAN[3], Oluwafolake OJO[4], Lawrence Omotosho[5]
[1,2,3,5] Department of Computer Science, Osun State University, Osogbo, Osun State, Nigeria.
[4]Department of Information Technology, Osun State University, Osogbo, Osun State, Nigeria
. [1]ibraheem.ogundoyin@uniosun.edu.ng, [2]dimple.ogunbiyi@uniosun.edu.ng,
[3]john.adeniran@uniosun.edu.ng, [4]oluwafolake.ojo@uniosun.edu.ng,
[5]lawrence.omotsho@uniosun.edu.ng

Keywords: Sentiment analysis, movie reviews, convolutional neural networks, natural language processing, machine learning

Abstract – *Sentiment analysis is an automated technique used to determine customers' opinions regarding products or services. Despite existing advancements, current models for analysing movie sentiments require further refinement to achieve more accurate and reliable sentiment classification. The study therefore, presents an improved movie review sentiment analysis model, designed based on Convolution Neural Network (CNN) architecture and optimized using Term Frequency-Inverse Document Frequency (tf-idf) algorithm. The performance evaluation of the optimized sentiment analysis model was carried out through comparative analysis with a conventional CNN model, using performance metrics, including accuracy, precision, recall, and F1-score to assess model performance. The simulation was driven by a robust movie dataset sourced from Kaggle. This result suggests that Opt-CNN is better in term of reliability for sentiment classification, thereby offering enhanced reliability*

1. INTRODUCTION

Sentiment analysis is a natural language processing (NLP) technique used to determine the emotional tone or attitude expressed in a text [1]. Sentiment analysis categorises the sentiment as positive, negative, or neutral by analysing words, phrases, and context [1]. Sentiment analysis is widely used in social media monitoring, customer feedback analysis, and market research to gauge public opinion and understand customer sentiment [2]. The process typically involves using machine learning algorithms or lexicon-based approaches to identify and quantify the sentiment expressed, enabling businesses to make data-driven decisions based on public mood and opinion ([1];[3]). In the context of online movie reviews, sentiment analysis offers valuable insights into the overall reception and public opinion of the movies. It helps movie enthusiasts decide which films to watch and assists producers and distributors in understanding audience sentiments toward specific aspects, such as plot, acting, or visual effects. This information can guide marketing strategies and improve the quality and success of future movie releases [5].Several works have been reported in the literature applying different machine-learning models to sentiment analysis and reviews ([6]; [7]; [8]). Different approaches and techniques exist in designing and implementing sentiment classification model for online product reviews [6]. Sentiment analysis often uses machine learning, lexicons, or hybrid approaches [8]. Although domain-dependent, machine learning is widely used, due to its high classification accuracy for sentiment classification [7]. The lexicon-based method assesses word semantics as positive or negative using opinion lexicons, but relies heavily on comprehensive linguistic resources, which may be scarce for non-English language datasets. The hybrid approach combines machine learning and lexicon-based methods [7]. Sentiment analysis also face challenges such as cultural factors, sentence

negation, sarcasm, terseness, language ambiguity, and differing contexts, all of which complicate converting text into simple sentiments, which affect the reliability of the sentiment analysis [7].
However, few works have been reported in movie sentiment reviews, with challenges in determining accurate and reliable models for implementing movie sentiment classification [7]. The reliability of sentiment analysis is faced by many obstacles, including cultural subtleties, negated statements, sarcastic language, concise expressions, ambiguity and contextual differences, which conventional machine and deep learning may not capture properly, highlighting the need for advanced optimized models and techniques.
To address these challenges of accuracy and reliability in sentiment analysis and classification, this study proposed an optimised machine learning model for implementing online movie sentiment reviews, aiming to enhance the effectiveness of sentiment analysis in detecting the true sentiments expressed by users.

2. RELATED WORKS

Several works have been reported in the literature on sentiment analysis.
Vidyabharathi et al. in [9] adopted RNN, LSTM, BiLSTM and GRU models in an experimental procedure to investigate an effective sentiment analysis of social media expressions. They generated model using Keras and Tensorflow libraries. The model's prediction operations were applied to an experimental dataset. The study was able to determine whether a tweet or expression on social media is positive or negative sentiments.
Sobia et al. in [10] carried out a study on Amazon product sentiment review data using machine learning techniques. The study's primary aim was to analyse sentiments contained in consumers' expressions so as to grasp their preferences better. The study obtained dataset, rating opinion that was detected initially in each analysis. The analysis involved performing pre-processing operations such as tokenization, stemming, and stopword removal from the dataset to extract meaningful reviews that are either positive or negative. The results from experiment showed that user comments can show the behaviour of a buyers' level of satisfaction. Also, it has been proved that the developed system can give reports for any product sold on the Amazon website
Aru in [11] analysed and visualized social networks sentiment data using Machine learning algorithms. The aim of the paper was to investigate the application of machine learning techniques to sentiment classification problem, compared with statistical approach. The study used Keras, Scikit learn and PyTorch Python libraries. Performance evaluation of the models was carried out using, widely used performance metrics, such as accuracy, precision, confusion matrix, recall, and F score. LSTM and BERT models showed accuracy, at about 0.8. However, the BERT performed better in precision, recall, and F score.
Vimala et al. in [12] studied a deep learning approach to predicting products' sentiment ratings in a comparative analysis study. The study aimed to use deep learning models to predict customers' review ratings using an e-commerce dataset containing women clothing reviews. Data augmentation using the easy data augmentation approach was applied on the dataset to increase the amount of data, which resulted in two versions of datasets. The models were evaluated and findings showed that the Neural Network-based using Word2Vec achieved the highest accuracy (96%) and F-score (91.1%). RNN emerged as the best individual model, with accuracy and F-score rates of 87.5% and 83.5%, respectively.
Dang in [13] performed a sentiment analysis based on Deep Learning in a comparative study. The study aimed to study the performance of three widely used deep learning techniques including; DNN, CNN, and RNN, performed on eight datasets. In the pre-processing stage, data cleaning and feature extraction were performed. Various deep-learning models were utilized during the training stage. The result from experiments showed that CNN outperformed the other models considered, in terms of accuracy and CPU runtime. In terms of reliability RNN performed slightly better than CNN but computational time was more.
Zhu et al. in [14] investigated an "interactive dual Attention Network for Text Sentiment Classification". The study proposed a model called Interactive Dual Attention Network (IDAN), with the intention to determine the interconnection between contextual meaning and sentiment directed information for sentiment

classification. The study designed an algorithm which combined various features attributing sentimental information from text. The contextual word representation, determined through the BERT pre-training model was used as the embedding layer of IDAN. Also, the study adopted two Bidirectional LSTM (BiLSTM) networks to learn the dependencies on contextual meaning and sentiment directed information. Result analysis demonstrated the effectiveness of the proposed method.

In [15], Wenkuan et al. in their work “an improved approach for text sentiment classification based on a Deep Neural Network via a Sentiment Attention Mechanism”, proposed the Sentiment-feature-enhanced Deep Neural Network (SDNN) for sentiment classification in text tasks by incorporating sentiment knowledge into the deep neural network architecture, through a sentiment attention method to learn sentiment enhanced word representations. The results from experiment showed that the SDNN has better performance than some other methods for sentiment classification tasks.

Dashtipour et al. in [16] presented Sentiment Analysis of Persian Movie Reviews Using Deep Learning, aiming to address the gap in sentiment analysis research for non-English languages, specifically Persian. The study applied deep learning algorithms, including Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, to classify Persian movie reviews. Where, LSTM models outperformed CNNs, demonstrating better capability in capturing the sequential nature of language. However, a notable limitation is the study's narrow focus on Persian, which may limit the generalizability of the findings to other languages or contexts

In [17], Nagaraj et al. presented movie reviews using sentiment analysis. The study explored the application of natural language processing (NLP) techniques in analysing and classifying sentiments in movie reviews. The scope of th/e work included obtaining dataset of movie reviews, preprocessing the data, and applying machine learning algorithms to train a sentiment classification model. Besides, comprehensive approach, which involved not just the use of conventional sentiment analysis methods but also the evaluation of the model’s performance using various metrics such as accuracy, precision, recall, and F1 score. However, the study's limitation is its reliance on conventional machine learning models, which usually struggle with complex expressive features such as sarcasm, cultural nuances, and context-specific meanings, thus affecting the overall reliability of sentiment classification.

The above reviewed literatures have contributed immensely to sentiment analysis and reviews. However, few works have been reported in movie sentiment reviews, with challenges in determining accurate and reliable models for implementing movie sentiment reviews. Therefore, this study proposed a reliable model for implementing online movie sentiment reviews.

3. RESEARCH METHODOLOGY

The movie review sentiment analysis model was designed using an optimised Convolution Neural Network (Opt-CNN). The model was simulated in Python version 3.12.3. The proposed Opt-CNN model performance was compared with conventional CNN model using performance metrics, such as accuracy, precision, recall, and F1-score metrics. The methods include data acquisition, text preparation, sentiment detection, sentiment classification, and output presentation.

3.1. Model Architecture and Design

Figure 1 illustrates the proposed online movie classification model architecture. The architecture design begins with data collection module, sourcing a diverse and comprehensive dataset of online movie reviews from Kaggle's repository. This rich dataset facilitates robust training and testing of the proposed movie reviews sentiment classification model. The subsequent feature extraction and text preparation module plays a critical role in converting raw text into a suitable format for analysis. Key pre-processing techniques employed include, tokenization, splitting text into individual words or tokens, stop word removal, elimination of common words ("and," "the," "is") that do not contribute to sentiment classification. These pre-processing steps reduce noise, enhance meaningful features, and optimize text for sentiment analysis. The sentiment analysis and classification module leveraged Convolution Neural Network (CNN) optimized with Term Frequency-Inverse Document Frequency (tf-idf algorithm. tf-idf transforms textual data into numerical features, capturing important words within the dataset context. Figure 2 provides a detailed illustration of the optimized CNN (Opt-CNN) architecture designed specifically for the movie review sentiment analysis and classification. It consists of input layer where the movie reviews dataset will be fed into the model, the layer also performs embedding function. Next, is the convolution layer, which scans and filters the input dataset both vertically and horizontally. The pooling layer reduces feature map dimensionality, while dense layer, also known as the fully connected layer performs functions of conventional neural network. The output layer will provide the output, in the case of this study, the movie reviews sentiment classification.

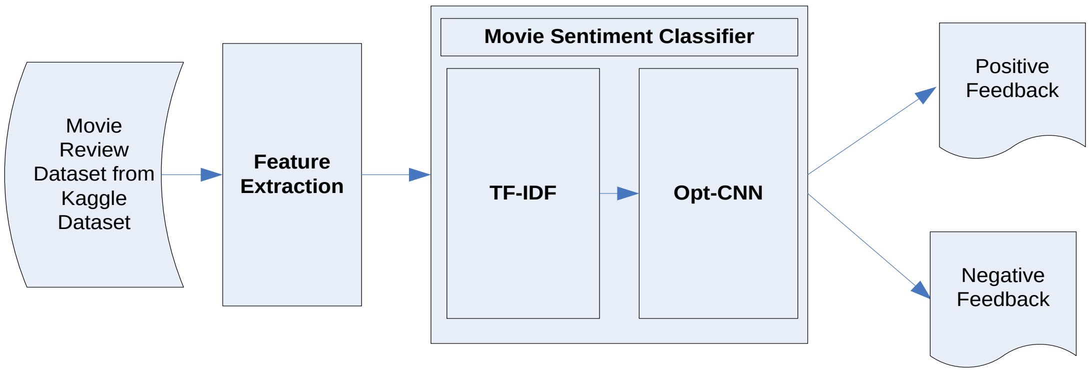

Figure 1: Architecture of the proposed online movie review sentiment classification model

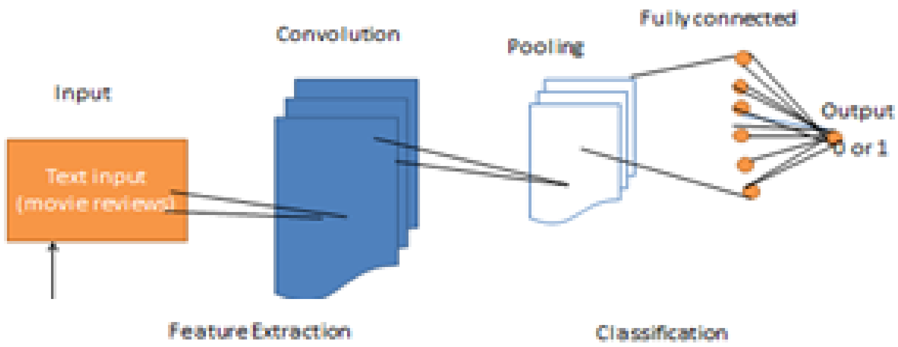

Figure 2: CNN Architecture for the Proposed online movie review sentiment classification

3.2. Dataset Description

The dataset used in this research was sourced from Kaggle, an online repository and a reputable platform for curated and structured datasets. The movie review dataset was selected from Kaggle due to its comprehensive metadata, ensuring data quality and consistency for analysis.

This approach eliminated the need for web scraping, often resulting in noisy and unstructured data. The dataset is presented in a text format, specifically in CSV (Comma Separated Values) format. It comprises 50,000 Internet Movie Database (IMDB) movie reviews, divided into training and test datasets. 80% of the dataset was for training, consisting of 40,000 reviews, while the remaining 20% split was for model testing, containing 10,000 reviews. Upon examination, the movie reviews dataset for sentiment analysis revealed three primary columns: movie title, review, and rating. The movie title column contains the name of the film being reviewed. This information enables the categorisation and organisation of reviews by specific titles, facilitating the aggregation of sentiment data at the film level and performing title-specific analysis. The review column comprises the textual content of user reviews. This text serves as the primary data source for the sentiment analysis model training, where

natural language processing techniques are applied to classify sentiment as either positive or negative.
The rating column quantitatively measures the reviewer's opinion through a numerical score or rating scale of 1 to 10. This numerical rating provides a straightforward sentiment assessment to calibrate or validate the sentiment analysis on the review text. Combining these columns allows sentiment scores derived from text to be correlated with numerical ratings. This integration offers a nuanced understanding of how textual sentiment aligns with rating scales, enhancing the robustness of sentiment analysis by incorporating both 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

3.3. Model Formulation

The optimised movies review sentiment analysis model embeds tf-idf in the CNN algorithm to form Opt-CNN. The model captured numerous essential stages and factors mathematically represented through various CNN layers. The mathematical description of Convolutional Neural Network (CNN), which takes movie reviews as input, analyses the reviews, generates the sentiments in the reviews and classifies them based on their polarity, either positive or negative, to get feedback for movie improvement, is presented as follows:

i. Mathematical Description of CNN Model Convolutional Layer:

Let x be the input feature map with size H x W (height x width), containing the movie reviews dataset from Kaggle. Then,

$$y\,[i,j] = \sigma\,(\Sigma\Sigma W[k,1] * x[i+k, j+i] + b)$$
(1)

Where:
i and j are the spatial coordinates (row and column indices) of the output feature map of the sentiment classification.
k and l are the spatial coordinates (row and column indices) of the CNN filter/kernel.
The convolutional layer applies a filter/kernel to the input feature map, scanning horizontally and vertically. For each position (i, j), the layer computes the dot product of the filter weights $W[k, 1]$ and the corresponding input values $x[i+k, j+i]$, 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 downsampling by applying a max pooling operation to non-overlapping regions of size K x K in the input feature map of the movie reviews dataset. This reduces spatial dimensions while retaining important features. This is given in equation (2) as follows

$$y\,[i,j] = \text{pooling}(x[i{:}\, i+k, j{:}\, j+k])$$
(2)

Where:
i and j are the spatial coordinates (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.

Activation Function:

The activation function adapted is the Rectified Linear Unit (ReLU). The activation function is given in equation (3) below:

$$\sigma\,(x) = \max(0, x) \qquad (3)$$

where:

σ is the activation function.

Fully Connected Layer:
Fully connected layers are also known as Dense or Multilayer Perceptron (MLP) layers given as.

$$y = \sigma(W * x + b) \quad (4)$$

Where:
y, is the output, representing the sentiment classification.
W, is the weight matrix (filter/kernel) with size K x K (number of rows x number of columns.
b, is the bias term.

ii Mathematical Description of TF-IDF Optimisation
The Term Frequency-Inverse Document Frequency (tf-idf) algorithm transforms the movie review textual data into numerical features, capturing the importance of words in the context of the dataset used. The mathematical description of tf-idf optimisation is presented as follows:

$$\mathrm{tf-idf}\,(w, d, D) = \mathrm{tf}(w, d) \times \mathrm{idf}(w, D) \quad (5)$$

Where:
$\mathrm{tf-idf}\,(w, d, D)$ is the $\mathrm{tf-idf}$ value for word w in document d and corpus D
$\mathrm{tf}(w, d)$ is the term frequency of word w in document d
$\mathrm{idf}(w, D)$ is the inverse document frequency of word w in corpus D

Term Frequency (tf):

$$\mathrm{tf}(w, d) = \frac{\mathrm{f}(w,d)}{|d|} \quad (6)$$

Where:
$\mathrm{f}(w, d)$ is the frequency of word w in document d
$|d|$ is the total number of words in document d

Inverse Document Frequency (IDF):

$$\mathrm{IDF}(w, D) = log\,\left(\frac{|D|}{|\{d \in D : w \in d\}|}\right) \quad (7)$$

Where:
$|D|$is the total number of documents in Corpus D
$|\{d \in D : w \in d\}|$ is the number of documents in D that contain the word w
log is the natural logarithm

Therefore, the combined $\mathrm{tf-idf}$ equation is:

$$\mathrm{tf-idf}\,(w, d, D) = \left(\frac{\mathrm{f}(w,d)}{|d|}\right) \quad (8)$$

Equation (8) calculates the $\mathrm{tf-idf}$ value for each word in each document, considering both the frequency of the word in the document and its rarity across the entire corpus. Therefore;

$$\mathrm{tf-idf}\,(x) = \left(\frac{\mathrm{f}(w,d)}{|x|}\right) \times\ log\left(\frac{|D|}{|\{d \in D : w \in d\}|}\right) \quad (9)$$

iii. Mathematical Description of the Optimized Convolutional Neural Network with tf-idf (Opt-CNN)
The proposed movie review sentiment classification model was formulated based on a Convolutional Neural Network and optimised with the tf-idf technique (Opt-CNN). The model takes text data (movie reviews) as input. The combined mathematical equation showing how tf-idf was used to optimise CNN is presented as follows: If equations (4) and (9) are combined, then equation (10) is formed

$$y = \sigma(W \times \mathrm{tf-idf}\,(x) + b) \quad (10)$$

From (10), $\mathrm{tf-idf}$ first takes input text data x, then is later used as input to the CNN model. The weight matrix W and bias term b are learned during training to optimise the model's performance. The tf-idf representation $\mathrm{TF-IDF}(x)$ can be calculated using the equation (9) as earlier presented; this is written in terms of input x as follows:

$$\mathrm{tf-idf}\,(x) = \left(\frac{\mathrm{f}(s,\ x)}{|x|}\right) \times\ log\left(\frac{|D|}{|\{d \in D : s \in d\}|}\right) \quad (11)$$

where s is a word in the vocabulary, x is the input text data, $\mathrm{f}(s,\ x)$ is the frequency of word s in x, $|x|$ is the total number of words in x, D is the corpus, and $|D|$ is the total number of documents in D.
Using $\mathrm{tf-idf}$ to optimise the CNN model, we can improve the model’s ability to capture relevant features from the text data and reduce the impact of irrelevant or redundant features. Therefore, the CNN Model with $\mathrm{tf-idf}$ optimisation:

Input Layer:
$x \in \mathbb{R}^V$ (one-hot encoded text vector) (12)
V is the vocabulary size
$e = \mathrm{tf-idf}\,(x) \in \mathbb{R}^V$ ($\mathrm{tf-idf}$ transformed vector) (13)

Convolutional Layer:
$c = \sigma(W\_c * e + b\_c)$ (14)

where:
W_c is the convolutional weight matrix, and b_c is the convolutional bias term.

Pooling Layer:
p = pooling (c) (15)

Fully Connected Layer:
$y = \sigma(W\_f * p + b\_f)$ (16)

Where:
W_f is the fully connected weight matrix, and b_f is the fully connected bias term.

Combining the TF-IDF optimisation and CNN model:
$y = \sigma(W\_f * pooling\ (\sigma(W_c * \mathrm{tf-idf}\,(x) + b_c)) + b\_f)$ (17)

Where:
$\mathrm{tf-idf}\,(x)$ is the $\mathrm{tf-idf}$ transformed input vector data movie review data.
Equation (17) integrates $\mathrm{tf-idf}$ into the CNN architecture, allowing the model to leverage the importance of each term in the text data. The pseudocode of the Opt-CNN model is presented in algorithm 1, and the architecture is presented in Figure 2

From the pseudocode, convolutional_layer (x, W) applies the convolutional operation to the movie reviews dataset, as input x with weights W. Pooling_layer (x) : apply pooling operation to input x. Fully_connected_layer (x, W, b) applies a fully connected operation to input x with weights W and bias b.

3.4. Model Simulation

Simulation is an evaluation technique which represents the behavior of a real life model, in reference to time, frequency etc., In this study, simulation technique was employed to carry out the performance evaluation of the proposed Opt-CNN by measuring its performance relative to conventional CNN based on the selected metrics of accuracy, precision, recall, and f1 score. The classification of the movie reviews sentiment was carried out using the model descriptions of Opt-CNN and CNN in section 3.3 and algorithm in section 3.4. The process included steps such as text cleaning, tokenisation, and model training, leading to the creation of predictive models that can classify the movie reviews sentiments dataset. The simulation was carried out in a Python environment with frameworks such as Pandas, NumPy, Seaborn, Matplotlib, Scikit-learn, Keras, and TensorFlow.
The cleaned and tokenized movie reviews sentiment dataset obtained from kaggle were used to train the sentiment classification models (Opt-CNN and CNN). First, the movie review dataset were divided into training and testing datasets. This step ensured that the proposed Opt-CNN model performances could be evaluated on unseen test data. Following the data splitting, the Opt-CNN and CNN models were created using Keras. For instance, the models (Opt-CNN and CNN) include the input or an embedding layer for word embeddings, CNN layers for capturing sequential dependencies, dropout layers for regularization, and a dense layer for the output. In addition, the Opt-CNN has the tf-idf as optimiser component. The next step was model training, which involved fitting the models to the training data. The step involves training both the proposed model Opt-CNN) and existing model (CNN) for a specified number of epochs, with a batch size, and validating their performance on the test set. The models, during training and testing were assessed for performance on the test dataset using selected metrics, such as accuracy, precision, recall, and F1-score.
Figure 3 is the simulation scenario created and adopted for the study experimental procedure.

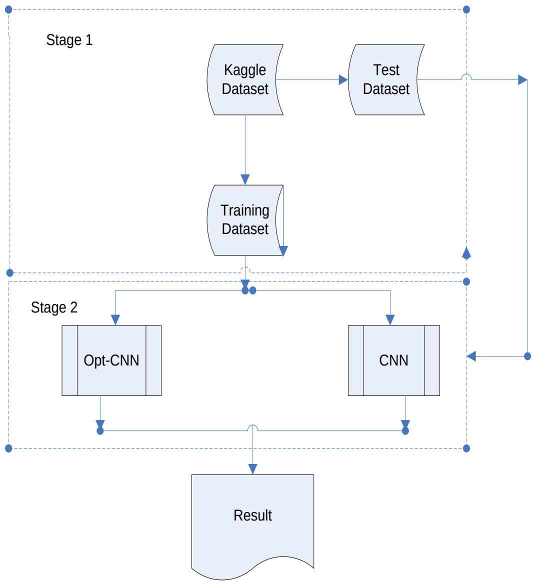

Figure 3: Simulation Diagram Model for Movie Review Sentiment Classification

This process includes text cleaning, tokenisation, model training, and creating a predictive model to classify sentiments in the input text dataset. The simulation was conducted in Python using frameworks such as Pandas, NumPy, Seaborn, Matplotlib, Scikit-learn, Keras, and TensorFlow.

The Opt-CNN model classified sentiment into positive or negative. The classification is done using the algorithm depicted in Algorithm 1.

Algorithm 1: Pseudocode of the Opt-CNN model

```
// Input Parameters
 x = input text data
D = corpus (collection of documents)
V = vocabulary (set of unique words)
W = weight matrix of CNN
b = bias term
f = activation function (ReLU)
num_epochs = number of training epochs
batch_size = batch size for training

// tf-idf Calculation
function tf − idf (x) :
  for each word w in x:
  tf = frequency of w in x /total number of words in x
    idf = log(|D| / |{d ∈ D: w ∈ d}|)
    tf_idf = tf ∗ idf
  return TF − IDF representation of x

// CNN Model
function CNN(x):
  x = tf − idf (x)
  convolved = convolutional_layer(x, W)

  pooled = pooling_layer(convolved)
  flattened = flatten(pooled)
  output = fully_connected_layer(flattened, W, b)
  return f(output)

// Training Loop
for epoch = 1 to num_epochs:
  for batch in batches(D, batch_size):
    x_batch = batch
    y_batch = labels(batch)
    output_batch = CNN(x_batch)
    loss = calculate_loss(output_batch, y_batch)
    update_weights(W, b, loss)
    update_bias(b, loss)

// Testing Loop
for x_test in test_data:
  output = CNN(x_test)
  predicted_label = argmax(output)
  return predicted_label
```

3.5. Model Performance Evaluation

The model was evaluated using metrics which include:

i. Precision: shows the ratio of the right review to all the time's review classification is right

$$Precission = \frac{TP}{TP + FP} \quad (18)$$

ii. Recall: shows how good the model is at finding the right review

$$Recall = \frac{TP}{TP + FN} \quad (19)$$

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

$$\mathrm{F1-score} = \frac{2*(Precision*Recall)}{Precision+Recall} \quad (20)$$

iv. Accuracy, which tells us how often the model is right compared to the real review

$$Accuracy = \frac{TP + TN}{TP + FN+FP+FN} \quad (21)$$

Where:

TP = True positive, FP = False positive, TN = True negative, FN = False negative

4. RESULTS AND DISCUSSION

This section presents and extensively discusses the simulation results of the proposed Opt-CNN-based movie reviews sentiment classification model compared with conventional CNN.

4.1. Results

In this study, the proposed Opt-CNN based sentiment classification for online movie reviews and the CNN based sentiment classification for online movie reviews were simulated, the simulation results were compared. The simulation was conducted on both training and test datasets. The performance evaluation of the two models (Opt-CNN and CNN) based on metrics defined in section 3.9 are presented in Tables 2 and 3. Table 2 is the performance of the Opt-CNN model during training as compared with CNN based model.

Table 2, shows the performance evaluation of Opt-CNN-based movie reviews sentiment classification model as compared with CNN-based model during training. Opt-CNN had accuracy 0.9147, precision of 0.9044, recall of 0.9242, f1-score of 0.9185 and AUC-ROC of 0.9602. But the conventional CNN based movie reviews sentiment classification model had accuracy 0.9085, precision of 0.9027, recall of 0.9107, f1-score of 0.9123 and AUC-ROC of 0.9474. During testing period, as presented in Table 3, Opt-CNN had accuracy 0.8747, precision of 0.8447, recall of 0.8934, f1-score of 0.8858 and AUC-ROC of 0.9402. But, CNN had accuracy 0.8485, precision of 0.8247, recall of 0.8880, f1-score of 0.8552 and AUC-ROC of 0.9274.

Table 2: Performance evaluation of Opt-CNN-based movie reviews sentiment classification model as compared with CNN-based model during Training

S/N	Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
1	Opt-CNN	0.9147	0.9044	0.9242	0.9185	0.9602
2	CNN	0.9085	0.9027	0.9107	0.9123	0.9474

Table 3: Performance evaluation of Opt-CNN-based movie reviews sentiment classification model as compared with CNN-based model during Testing

S/N	Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
1	Opt-CNN	0.8747	0.8447	0.8934	0.8858	0.9402
2	CNN	0.8485	0.8247	0.8880	0.8552	0.9274

4.2. Discussion of Result

The simulation results showed improved accuracy and reliability of the proposed Opt-CNN-based movie reviews sentiment classification model. The improved performance of the Opt-CNN is due to its optimisation using the tf-idf optimisation technique. The tf-idf transformed text data into numerical vectors, enabling CNN to process the movie review text data. Also, the tf-idf extracted relevant features from the movie review dataset, highlighting important words, before getting to the convolutional layer which will in turn extract again more important features from already filtered features by tf-idf. More importantly, the opt-CNN had better performance because the tf-idf used as the optimiser reduced the dimensionality of the movie reviews dataset, thereby decreasing the impact of noise and irrelevant features. The output from tf-idf served as input to the CNN model, which later performed the sentiment classification. The tf-idf has cleaned the data and reduced its size.

CNN model will work on. This was responsible for the better performance of the Opt-CNN compared to conventional CNN, which takes the raw data without the initial pre-processing performed by tf-idf.

The better performance of Opt-CNN-based movie review sentiment classification on the training dataset was mainly because the model had learnt the data pattern during training. It was easy for the model to classify the dataset used in training it better. The performance on the test dataset was the reverse of the training dataset. The model had not been exposed to the dataset for testing, the reason for the drop in performance. The Opt-CNN demonstrated a higher recall and AUC-ROC value, making it a better model for effective sentiment classification in this study. It also exhibited a shorter running time, indicating higher computational efficiency. At the same time, CNN showed a balanced performance. The Opt-CNN's ability to correctly identify positive sentiments with fewer false negatives resulted from optimisation, improving sentiment classification tasks.

5. CONCLUSION

The study successfully developed a movie review sentiment classification model using CNN optimised with the TF-IDF algorithm (Opt-CNN). The Opt-CNN outperformed the CNN model in terms of recall and overall performance. The Opt-CNN model's ability to achieve a higher recall of 0.9044 % in training and 0.8447% in testing indicated its effectiveness in sentiments and classification, which is crucial for understanding customer feedback and enhancing decision-making processes. Future research should consider incorporating additional pre-processing techniques and extending the analysis to larger datasets or domains. Additionally, hybrid models combining CNNs with machine-learning approaches could improve sentiment classification accuracy and robustness.

6. ACKNOWLEDGMENT

The authors acknowledge the support of Professor C.O. Akanbi and Professor P.A. Idowu for their technical and editorial support.

7. 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, 1008, 012011., 2018. https://doi.org/10.1088/1742-6596/1008/1/012011.

[2] Nandwani, P., and Verma, R., "*A review on sentiment analysis and emotion detection from text*", Social Network Analysis and Mining, 11(1), 1-19., 2021.https://doi.org/10.1007/s13278-021-00776-6.

[3]Shathik, J. A. and Krishna P. K. "*A Literature Review on Application of Sentiment Analysis Using Machine Learning Techniques",* International Journal of Applied Engineering and Management Letters (IJAEML), 4(2), 41-67, 2020. DOI: http://doi.org/10.5281/zenodo

[4] Drus, Z., and Khalid, H. *"Sentiment analysis in social media and its application: Systematic literature review"*, Procedia Computer Science, 161, 707–714,2019 https://doi.org/10.1016/j.procs.2019.11.174

[5] Kalaivani, P., "*Sentiment Classification of movie reviews by supervised machine learning approaches"*, Indian Journal of Computer Science and Engineering (IJCSE), *4*(4), 285–292, 2013.

[6] Bordoloi, M., and Biswas, S. K., "*Sentiment analysis: A survey on design framework, applications and future scopes*", Artificial intelligence review, *56*(11), 12505-12560, 2023

[7] Chandra, R. and Vaddatti, B. "*Sentiment Analysis using Deep Belief Network for User Rating Classification*", International Journal of Innovative Technology and Exploring Engineering, *10 (8),* 87-91, 2021

[8] 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* (pp. 35–48). IGI Global., 2020 http://dx.doi.org/10.4018/978-1-5225-8575-6.ch003.

[9] 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, *2 (10),* 4898-4908, 2021

[10] Sobia, W., Xi, C., Tian, S., Muhammad, W. and Jhanjhie, N. Z., *"Amazon Product Sentiment Analysis using Machine Learning Techniques"* Revista Argentina de Clínica Psicológica*, 30, (1),* 695-703, 2021

[11] Aru, O., "*Sentiment analysis and visualisation of data from social networks using Machine learning algorithms".* Dissertation submitted to the Department of Computer Science in partial fulfilment of the requirements for the degree of Master of Science in Computer Science, 2021

[12] 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*" Journal of Supercomputer, 78, 7206–7226, 2022. https://doi.org/10.1007/s11227-021-04169-6

[13] Dang, N. C., Moreno-García, M. N. and Prieta, F. D. *Sentiment Analysis Based on Deep Learning: A Comparative Study"* Electronics, *9 (483),* 1-29, 2020

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

[15] Wenkuan, L., Peiyu L., Qiuyue Z. and Wenfeng, L. *"An Improved Approach for Text Sentiment Classification Based on a Deep Neural Network via a Sentiment Attention Mechanism"*, Future Internet, 11, (96), 1-15, 2019

[16] 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

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