OPEN ACCESS
ISSN 2460-9056
journals.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 10, Issue. 1, August 2025. pp. 38-53
doi:10.21108/indojc.v10i1.9687

Sentiment Analysis of the Mobile Legends: Bang Bang Application Using a Hybrid CNN-LSTM Model

Eric Nur Rahman[1], Yuliant Sibaroni[2]

School of Computing, Telkom University
Telecommunication Street No. 1, Bandung 40257, Indonesia

[1] ericnur@student.telkomuniversity.ac.id
[2] yuliant@telkomuniversity.ac.id

Abstract

The increasing number of user reviews on the Google Play Store is a challenge in understanding user opinions and experiences with apps. One of the most discussed apps is Mobile Legends: Bang Bang (MLBB), a popular game with millions of downloads and reviews from Indonesian users. The problem faced is the limitation of conventional sentiment analysis models in understanding sentences and context simultaneously, making it less than optimal in analyzing user sentiment. This study proposed a comprehensive sentiment analysis system for MLBB application reviews, utilizing a hybrid CNN-LSTM architecture with a systematic optimization approach. A dataset comprising 30,000 balanced Indonesian user reviews was extracted from the Google Play Store using web scraping techniques and then processed through an extensive pre-processing pipeline, which included data cleaning, case folding, stopword removal, and stemming. Five experimental scenarios were conducted to optimize model performance through feature engineering and algorithmic enhancement. The baseline CNN-LSTM model achieved 71.97% accuracy, which was progressively improved through TF-IDF vectorization with optimal N-gram (1,2) configuration, max features optimization reaching 10,000 features, FastText embedding feature expansion using a 300-dimensional Indonesian pre-trained model, and optimizer selection experiments across five algorithms. The final optimized hybrid CNN-LSTM model, using the RMSprop, demonstrated a breakthrough performance of 88.84% accuracy with remarkable consistency (standard deviation of 0.000754), representing a 23.4% improvement over the baseline. This research contributes to the field of sentiment analysis, especially for game applications, by proving that a combined approach can produce a more accurate and reliable system for understanding user opinions.

Keywords: Classification, FastText, Hybrid CNN-LSTM, Sentiment Analysis, TF-IDF

Abstrak

Jumlah ulasan pengguna yang terus meningkat di Google Play Store menjadi tantangan tersendiri dalam memahami opini dan pengalaman pengguna terhadap aplikasi. Salah satu aplikasi yang banyak dibahas adalah Mobile Legends: Bang Bang (MLBB), gim populer dengan jutaan unduhan dan ulasan dari pengguna Indonesia. Permasalahan yang dihadapi adalah keterbatasan model analisis sentimen konvensional dalam memahami kalimat dan kontekstual secara bersamaan, sehingga kurang optimal dalam menganalisis sentimen pengguna. Penelitian ini mengusulkan sebuah sistem analisis sentimen yang komprehensif untuk ulasan aplikasi MLBB, memanfaatkan arsitektur hybrid CNN-LSTM dengan pendekatan optimasi sistematis. Dataset yang terdiri dari 30.000 ulasan pengguna Indonesia yang berimbang diekstraksi dari Google Play Store menggunakan teknik web scraping dan kemudian diproses melalui pipeline pra-pemrosesan yang

Received on January, 2025. Accepted on August, 2025

ekstensif, yang mencakup pembersihan data, case folding, stopword removal, dan stemming. Lima skenario eksperimental dilakukan untuk mengoptimalkan kinerja model melalui rekayasa fitur dan peningkatan algoritma. Model dasar CNN-LSTM mencapai akurasi 71,97%, yang secara progresif ditingkatkan melalui vektorisasi TF-IDF dengan konfigurasi N-gram (1,2) yang optimal, optimalisasi fitur maksimum mencapai 10.000 fitur, perluasan fitur penyematan FastText dengan menggunakan model pra-terlatih bahasa Indonesia 300 dimensi, dan eksperimen pemilihan optimizer di lima algoritma. Model hibrida CNN-LSTM yang dioptimalkan terakhir, menggunakan RMSprop, menunjukkan kinerja terobosan dengan akurasi 88,84% dengan konsistensi yang sangat baik (standar deviasi 0,000754), yang menunjukkan peningkatan 23,4% dibandingkan dengan model baseline. Penelitian ini memberikan kontribusi dalam bidang analisis sentimen, khususnya untuk aplikasi gim, dengan membuktikan bahwa pendekatan gabungan yang dapat menghasilkan sistem yang lebih akurat dan andal dalam memahami opini pengguna.

Kata Kunci: Klasifikasi, FastText, Hybrid CNN-LSTM, Analisis Sentimen, TF-IDF

I. INTRODUCTION

IN the era of digital transformation, mobile app distribution platforms have become a key ecosystem that connects developers and users on a global scale. Google Play Store, as the largest Android app marketplace with over 3.5 billion monthly active downloads, provides an invaluable repository of user review data to understand public perception of mobile apps [1]. Based on Mobile Operating System Market Share Data in Indonesia, there are 89.77% Android users and 10.12% iOS users which means that Play Store Market Place users are more than the App Store, on Google Play Store there is a column of user ratings and reviews for applications that are available [2]. User reviews on this platform serve not only as a feedback mechanism but also as a crucial source of information that can be analyzed to extract comprehensive insights into user sentiment.

Mobile Legends: Bang Bang (MLBB), the top-rated mobile games in Southeast Asia, with over 1 billion global downloads, has become a gaming phenomenon that generates intense discussions among users [3]. More than 30 million user reviews on Google Play Store reflect the game's popularity, covering a diverse spectrum of sentiments that range from appreciation of the innovative gameplay to criticism of technical aspects [3]. The app is a hot topic of discussion given its popularity and the large number of players in Indonesia. The complexity of sentiment in MLBB reviews reflects the dynamics of the competitive mobile gaming industry, where a deep understanding of user perceptions is a crucial factor in determining product sustainability and development. Precise analysis of the sentiment reviews can provide strategic insights for developers to optimize features, address bugs, and inform user engagement strategies.

Sentiment analysis is a branch of machine learning that has undergone a paradigmatic evolution from traditional rule-based approaches to complex deep learning architectures [4]. Conventional methodologies that rely on manual lexicons and feature engineering have proven to have limitations in capturing the semantic and contextual complexity of natural language text, especially in the gaming domain, which is rich in specific terminologies and expressions [5, 6]. The development of word embedding techniques, such as FastText, has enabled a more robust representation of distributional semantics; however, it still requires a neural network architecture capable of integrating local features and sequential dependencies simultaneously [7]. A key challenge in sentiment analysis lies in the model's ability to capture hierarchical correlations of word, phrase, and sentence levels while maintaining optimal generalizability.

Moreover, lately, deep learning algorithms have gained popularity and demonstrated high-quality performance in sentiment analysis, such as the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) [8]. Although CNN is popularly used for image data, it can also achieve excellent results with text classification tasks [9, 10]. CNN architecture can extract local features that are invariant and identify significant n-gram patterns in text, capturing semantic compositionality through multiple convolutional layers with various kernel sizes [11]. LSTM excels at handling sequence patterns in text data, such as examining words that appear before the current word and those that follow it to gain further insight into the context [9, 12]. LSTM demonstrates exceptional capability in understanding context-dependent sentiment shifts and handling variable-length sequences characteristic of natural language data [13]. By combining these two models, it can utilize the advantages of each to produce more accurate and in-depth sentiment analysis. The CNN-LSTM hybrid model represents an architectural innovation that aims to integrate the complementary strengths of both neural network paradigms to optimize sentiment analysis performance [14, 15]. The model offers excellent potential to improve sentiment analysis performance compared to traditional models. The CNN-LSTM hybrid built by Hanane Mohaouchane et al. obtained an accuracy of 87.27%, outperforming the comparison model, because the model enables the network to initially capture local features through the CNN, followed by learning global features using the LSTM [9].

Therefore, this study aims to develop an optimized hybrid CNN-LSTM-based sentiment analysis system with five scenarios for MLBB on Google Play Store, with the expectation of producing superior performance.

II. Literature Review

This research is based on related research that has already been done. As in the research conducted by Raihan and Setiawan [8], conducted aspect-based sentiment analysis on Twitter posts about Telkomsel products. This research aims to overcome the problem of vocabulary mismatch caused by the unique and abbreviated language used on Twitter, and to improve the accuracy of sentiment classification for each aspect by utilizing feature expansion using FastText word embedding combined with classification algorithms. Features are extracted from pre-processed tweets using the TF-IDF (Term Frequency-Inverse Document Frequency) technique. Feature expansion is done by FastText word embedding to identify semantically similar words, thus reducing the problem of vocabulary mismatch. Adding FastText-based similar words to the feature vector further improves performance. The best results used the IndoNews corpus with Top-5 similar features (F1-Score: 95.93%, an increase of 27.91%). In research [10], introduced and evaluated a Convolutional Neural Network (DCNN) architecture to model the semantic content of sentences. This research was conducted to investigate whether dynamic convolutional architectures can outperform existing methods in various language understanding tasks, especially in terms of semantic representation and classification accuracy. The proposed method outperforms both neural and non-neural baselines in fine-grained and binary sentiment classification. The method achieves substantial improvement, gaining an accuracy of 87.4% and reducing errors by more than 25% compared to the strongest n-gram-based baseline.

In addition, based on research [11], which has reviewed more than 150 deep learning models, covering a wide range of categories, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), to hybrid models, quantitatively compared the models on 16 widely used benchmark datasets, showing their performance in terms of metrics such as accuracy, F1-score, and mean reciprocal rank. CNN achieves strong results in news categorization and topic classification, with deep variance and character level driving accuracy close to or above 95%. For datasets where capturing sequence or hierarchical information in the text is critical, CNN-LSTM hybrid models or models that add attention provide additional advantages. Research conducted by Rehman et al., [14] developed a hybrid deep learning model that combines CNN and LSTM architectures to improve sentiment analysis performance on movie review datasets. The main objective of this research is to overcome the limitations of existing sentiment analysis methods by proposing a hybrid neural network that leverages the advantages of both CNN and LSTM models. While CNNs are effective at extracting local features from text, they struggle to capture long-term dependencies. LSTM networks, on the other hand, excel at learning sequence patterns and long-term dependencies. This method achieved an accuracy of up to 91% on the IMDB dataset, marking a significant performance improvement compared to traditional machine learning models and standard deep learning models. In comparative analysis, the hybrid model's F-measure score shows a 4–8% improvement over standalone models. This model demonstrates scalability and effectiveness in extracting complex relationships in text data, making it suitable for large-scale sentiment analysis tasks.

Furthermore, research conducted by Hossain et al., [15] analyzed restaurant review sentiment using a dataset containing 1,000 restaurant reviews collected through web scraping. Each review was manually labeled as positive (500 reviews) or negative (500 reviews). To demonstrate the effectiveness of the hybrid model (CNN-LSTM) compared to traditional approaches in classifying customer opinions as positive or negative, the model achieved a high training accuracy of 94.22%. The training loss reached 0.155, and the validation accuracy (as reported alongside the loss) was 72.2%. This approach demonstrates that the use of deep learning methods, particularly the hybrid CNN-LSTM model, yields better predictive performance than traditional models. Despite facing challenges due to the diversity of the Bangla language, the results show significant potential for practical applications. The CNN-LSTM hybrid method has proven effective for restaurant review sentiment analysis, offering higher accuracy and robust metrics.

Although various approaches have been used for sentiment analysis, most of the previous research still focuses on relatively simple general domains or without deep optimization processes. Rehman et al. (2019) used a CNN-LSTM model for movie reviews, but did not apply feature expansion or advanced optimization algorithm testing [14]. Hossain et al. (2020) compared CNN-LSTM with conventional models for restaurant reviews, but the scale of the data is limited and has not involved non-English languages [15]. Raihan & Setiawan (2022) adopted FastText for feature expansion on Twitter data, but have not combined it with hybrid architecture and optimization experiments [8]. Therefore, this research contributes to developing a CNN-LSTM-based sentiment analysis system that is incrementally optimized in the domain of Indonesian-language game app reviews, which has not been studied in depth. It also integrates TF-IDF technique, FastText feature expansion, and optimization algorithm selection in five experimental scenarios to improve the accuracy and stability of the model. This research was conducted with the aim of producing superior performance.

III. RESEARCH METHOD

This research develops a sentiment analysis system for the MLBB application using a hybrid CNN-LSTM model that incorporates feature extraction with TF-IDF and FastText-based feature expansion. The pipeline of the proposed system is illustrated as flowchart in Figure 1.

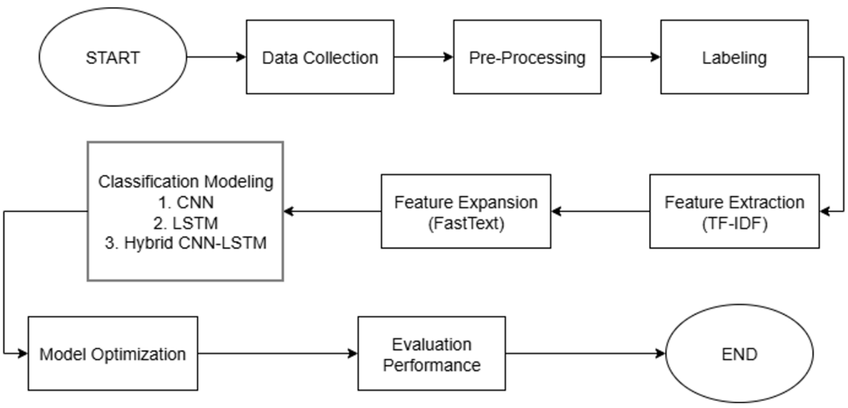

Fig. 1. Flowchart of The Proposed Model

A. Data Collection

The dataset in this research was retrieved through a web scraping process utilizing the Google Play Scraper library, with the package ID com.mobile.legends, given the dominance of the Android platform in Indonesia, which represents more than 90% of the market share. The scraping configuration focused on Indonesian-language reviews from users in Indonesia, with relevant rating filters to ensure a natural and unbiased sentiment distribution. A total of 11 key attributes were collected, including reviewId, userName, content, and score. The content attribute was used as text input for classification, and the score served as the ground truth in supervised learning. The entire dataset is saved in CSV format to facilitate pre-processing and further analysis. Figure 2 presents the crawling results in the form of a word cloud, illustrating the sentiments towards the MLBB game, with words such as "good", "fun", "great", and "cool" dominating the positive sentiments. At the same time, 'bad' and 'broken' appear in the negative sentiments, reflecting complaints about the game's performance and developers. This visualization provides an initial foundation for automatic sentiment analysis and subsequent evaluation of the recommendation system.

Fig. 2. Word Cloud of Positive (a) and Negative (b) Sentiments of Users towards MLBB

B. Data Pre-processing

This stage is important component in the sentiment analysis pipeline, aiming to optimize the quality of the input text before it is processed by the classification model. The process begins with the selection of relevant attributes from the original dataset. The content attribute serves as the primary feature for sentiment analysis, while the score is designated as the target variable. Evaluation of missing values showed good data integrity with no significant empty values. This stage includes [16]:

1) *Data Cleaning*
The data cleaning stage involves removing non-textual elements, such as URLs, hyperlinks, punctuation marks, numeric characters, and emojis.
2) *Case Folding*
Case folding is implemented by converting all text to lowercase to ensure consistency of lexical representation in the vocabulary.
3) *Stopword Removal*
Stopword removal is used to eliminate functional words with low semantic value, such as prepositions, conjunctions, and articles.
4) *Stemming*
Stemming implements to reduce morphological variations of words to their base form, which is essential for overcoming inflection in Indonesian and improving model generalization.
5) *Tokenization*
Tokenization breaks down sentences into smaller word units or tokens, which serve as an essential first step in text analysis to understand the linguistic structure and meaning of the data in more detail.

Table 1 presents an example of the final dataset after all pre-processing steps have been completed, including cleaning, case folding, and tokenization. This table presents the structured form of the data, which is ready for further analysis or model training.

TABLE I
Example of Final Dataset

ReviewID	UserName	Content	Score
d3ac7da0-3d81-43be-ac6e-459f25b0bbfe	Senia Pitaloka	servernya masalah setiap kali ada pembaruan “The server is having problems every time there is an update”	1

1dbfbd78-0ce3-4430-b0d9-69ae466a18a0	Nofry Mulia	pokoke mantap “the important thing is amazing”	5

C. Labeling

The data labeling process employs a binary classification scheme that transforms numerical ratings into categorical sentiment labels, where ratings 1-2 are classified as negative sentiment (label 0) and ratings 4-5 as positive sentiment (label 1), while rating 3 (neutral) is eliminated to avoid classification ambiguity. This binary approach was chosen considering that sentiment analysis in the gaming domain tends to be polarized, with users expressing clear satisfaction or dissatisfaction towards gameplay experience. A balanced sampling strategy was implemented to mitigate potential class imbalance using stratified random sampling with fixed random seed, setting the target at 15,000 samples for each sentiment class (30,000 total samples) to ensure experiment reproducibility and optimal binary classifier training [17]. Table 2 shows the labeling results.

TABLE II
EXAMPLE OF LABELED DATASET

ReviewID	UserName	Content	Score	Label
d3ac7da0-3d81-43be-ac6e-459f25b0bbfe	Senia Pitaloka	servernya masalah setiap kali ada pembaruan “The server is having problems every time there is an update”	1	0
1dbfbd78-0ce3-4430-b0d9-69ae466a18a0	Nofry Mulia	pokoke mantap “The important thing is amazing”	5	1

D. Feature Extraction TF-IDF

This process refers to the transformation of raw text into numerical formats that can be effectively interpreted and utilized by machine learning algorithms. This process includes techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) that capture different information about the context and meaning of words in the text [16, 18]. The goal is to capture linguistic and semantic patterns that can distinguish sentiments within a sentence or document. TF-IDF measures the significance of a word within a document by accounting for how often it appears in that document and how uncommon it is across the entire corpus. TF-IDF is calculated by giving a score that assesses the frequency of words and their rarity throughout the document [19]. The weight calculation formula in TF-IDF can be formulated below [19]:

$$(TF - IDF)_{i,j} = (TF)_i \times \log{(IDF)_j} \tag{1}$$

Where TF indicates a standard weight that indicates the frequency or associated frequency of the term, IDF indicates the global weight that shows the support of the term belonging to the set of tweets. Thus, $TF - IDF$ is used to transform text into a vector representation, where each word is assigned, a weight based on its importance within the document and the corpus [19].

E. Feature Expansion FastText

Feature expansion is a technique to enrich the representation of text data by adding additional semantic or contextual information to improve the model's understanding of word meaning [20]. FastText is an open-source NLP algorithm developed by Facebook AI Research that performs feature expansion by focusing on the internal structure of words through character n-grams, enabling efficient processing of out-of-vocabulary terms commonly found in gaming reviews [21]. Figure 3 illustrates FastText's workflow in converting text into vector representations for analysis purposes such as sentiment classification [22]. In this research, FastText converts pre-processed MLBB review text into dense vector representations by averaging word vectors and applying linear transformation to capture semantic relationships, particularly effective for handling Indonesian morphological variations and gaming terminologies that may not appear in standard training vocabularies [22].

This approach significantly enhances the model's ability to understand nuanced sentiments in gaming domain text, where specialized terms and informal expressions are prevalent [20].

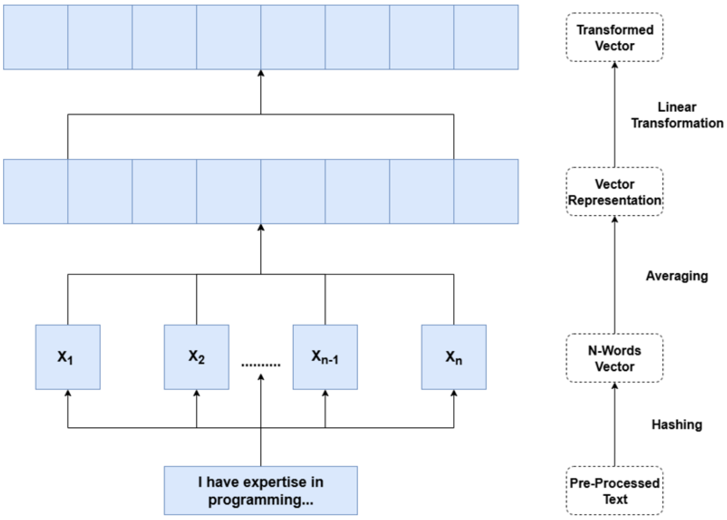

Fig. 3. FastText Architecture

F. Convolutional Neural Network (CNN)

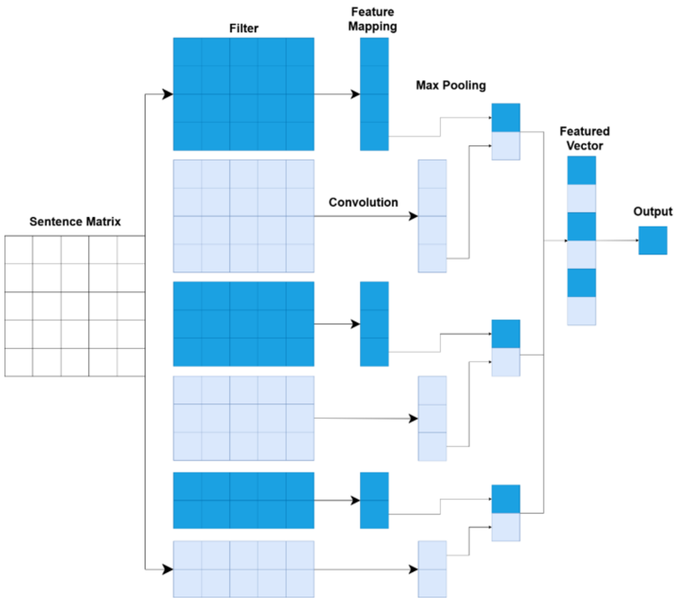

Fig. 4. CNN Architecture

CNN, initially conceived for image processing, has demonstrated strong performance in natural language processing applications, particularly sentiment analysis, attributable to its capacity to capture local patterns within text data [23, 24]. CNN works by applying convolutional filters on input data to extract local features such as key phrases or n-grams important for determining sentiment, followed by pooling layers to decrease dimensionality and highlight relevant information. Figure 4 illustrates the CNN workflow, where text is represented as a sentence matrix with embedded word vectors, and multiple filters are applied to perform convolution processes that result in feature maps highlighting essential sentiment patterns [25]. Each feature map undergoes max pooling to extract significant feature values, and the pooled results from all filters are combined into a feature vector representing the main sentence features, which is then passed to the output layer for classification.

$$h_i = f(\sum_{j=1}^{k} w_j \times x_{i+j-1} + b) \tag{2}$$

CNN in text processing where h_i is the convolution result at the i-th position, w_j is the kernel weight, x is input and f is the activation function. CNNs work by applying filters on a sliding basis to detect local patterns in text data, such as phrases or n-grams that contain specific sentiment meanings. In the context of sentiment analysis, CNN effectively extracts local features regardless of order, making it ideal for detecting keywords that appear in reviews.

G. Long Short-Term Memory (LSTM)

As a refined form of Recurrent Neural Network, LSTM is engineered to handle sequential data and effectively capture long-term dependencies within text [26, 27]. LSTM networks operate by utilizing memory cells and gate mechanisms (input, output, and forget) which organize the information flow. This permits the model to store or throw out information as needed throughout long sequences. This architectural design allows LSTMs to avoid the vanishing gradient issue that often hinders traditional RNNs, thereby enhancing their effectiveness for tasks where context and word order are critical, such as sentiment analysis. In sentiment analysis, LSTM excels at understanding the subtle relationships between words and phrases within a sentence or document, allowing the model to detect sentiment even in complex or lengthy texts accurately. LSTM models have shown improved performance over conventional machine learning, particularly in handling large datasets and domain-independent sentiment classification [27].

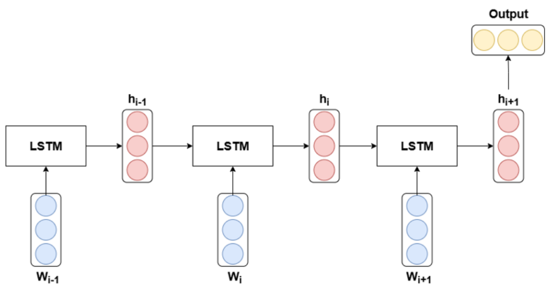

Fig. 5. LSTM Architecture

Figure 5 shows how LSTM works, where each word in the sentence (W_i) is represented in vector form and input sequentially into the LSTM units. Each LSTM unit processes the current input and the hidden state from the previous time step (h_{i-1}, h_i) to generate a new hidden state that incorporates historical context information. This memory mechanism enables the LSTM to capture long-term dependencies in the text, such as the relationship between distant words, which is crucial for comprehending the whole emotional meaning or opinion. After the entire set of words is processed, the final hidden state (h_{i+1}) is used to generate the final output in the form of sentiment prediction, be it positive, negative, or neutral. This approach is highly effective in handling complex and contextualized text sequences [28].

$$f_i = \sigma(X_f \,.\, [h_{i-1}, w_i] + b_f \tag{3}$$

$$k_i = \sigma(X_i \,.\, [h_{i-1}, w_i] + b_i \tag{4}$$

$$\tilde{C}_i = tanh(X_C \,.\, [h_{i-1}, w_i] + b_C \tag{5}$$

$$C_t = f_i * C_{t-1} + k_i * \tilde{C}_i \tag{6}$$

$$o_i = \sigma(X_o \cdot [h_{i-1}, w_i] + b_o \quad (7)$$
$$h_i = o_i * \tanh(C_t) \quad (8)$$

LSTM uses several formulas that govern which information is forgotten or retained through a gating mechanism. w_i is the current input, h_i is the hidden state, C_t is the cell state, and σ is the symbol of sigmoid. LSTM is effective in understanding long-term dependencies and sentence context, such as the relationship between widely separated words in text. LSTM is able to capture context that expresses negative sentiments in long and complex sentence structures. Overall, LSTM's ability to model sequence dependency and retain relevant context makes it a strong tool to extract sentiment from unstructured data.

H. Hybrid CNN-LSTM

CNN-LSTM is a hybrid architecture that integrates CNN with LSTM networks to utilize the advantages of both architectures in sentiment analysis [29]. This architecture enables the network to identify local features from a CNN and global features from an LSTM [9]. In this approach, a CNN layer is used first to extract local features and patterns from a sequence of word embeddings, such as key phrases or n-grams that may indicate sentiment. These extracted features are then passed to the LSTM layer, which is skilled at capturing long-term dependencies and contextual relationships across texts. This sequential processing allows the model for perceiving both the local and global structure of the language, making it highly effective for analyzing complex or lengthy texts [29, 30]. The main advantages of CNN-LSTM in sentiment analysis include higher accuracy, the ability to handle unstructured and noisy data, and superior performance over models that use only CNN or LSTM alone. This hybrid approach is particularly valuable for extracting meaningful insights from extensive user-generated content, including reviews and posts [31].

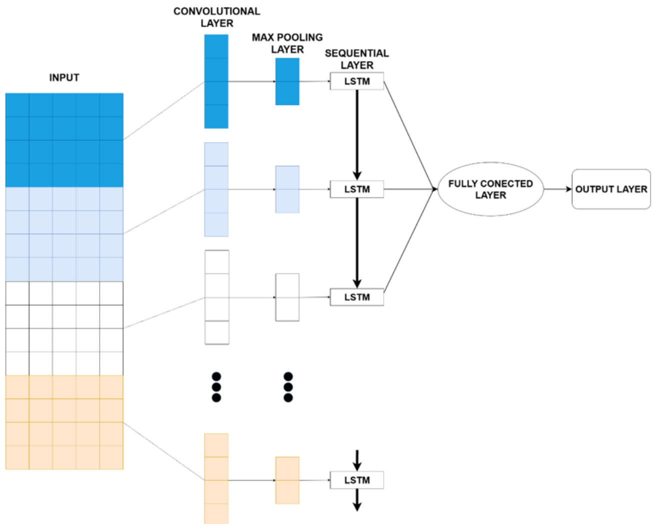

Fig. 6. Hybrid CNN-LSTM Architecture

Figure 6 illustrates the combined architecture of CNN and LSTM in sentiment analysis. The CNN-LSTM hybrid model combines the CNN formula for local feature extraction and the LSTM formula for contextual sequence processing. CNN first extracts important features from the text embedding, then these feature results are passed to LSTM which processes them with a memory formula. The process begins with input in the form of a text representation in matrix form. The CNN acts as an initial layer that extracts spatial features from the words in the sentence through convolutional filters, which are then processed by max pooling to reduce the dimensionality and capture the most salient essential features. Each of these pooled results reflects the local information of a particular section of text that contains sentiment elements, such as positive or negative phrases, or nuanced expressions. The results from the CNN are then passed to the LSTM layer, which works sequentially.

The LSTM here serves to capture the temporal relationship or sequence between features extracted by the CNN, allowing the model to understand the long-term context of the text. Each LSTM unit receives features from the pooling results and manages them through its internal memory, resulting in a deeper contextual representation. After the LSTM processes all the information, the results are consolidated in a fully connected layer for classification, and finally produce sentiment predictions in the output layer [32]. This combination utilizes the power of CNN in extracting essential features and the ability of LSTM in understanding sequences, making it highly effective for complex sentiment analysis [32, 33]. This combination allows the model to understand relevant keywords as well as the overall context of the sentence. In this study, the hybrid architecture can be used for reviews that often contain mixed statements, as it can assess the overall nuances of sentiment more accurately.

I. Model Optimization

In sentiment analysis using deep learning as a classification model, optimizers play a crucial role in training artificial neural networks efficiently and effectively [34, 35]. Each optimizer has unique characteristics in managing weight updates during the training process, so it is essential to understand its advantages and disadvantages according to the complexity of the text data being processed. The following is an explanation of each optimizer used in this research.

1) *Adam*
Adam optimizes the learning process by merging the benefits of momentum, adaptively adjusting the learning rate using the gradient moments.
2) *SGD*
SGD modifies the parameters by leveraging the gradient derived from the loss function; however, it is highly sensitive to the choice of learning rate and may converge slowly.
3) *RMSProp*
RMSProp adjusts the learning rate retrieved on most recent gradient, which helps in handling non-stationary objectives and has shown strong performance in sentiment analysis tasks.
4) *Adamax*
Adamax, a variant of Adam, utilizes an infinite norm (absolute maximum) for normalization, which provides stability in some instances.
5) *Adagrad*
Adagrad adjusts the learning rate based on the frequency of parameter updates, favoring sparse features, but it can suffer from rapid degradation of the learning rate.

J. Evaluation Performance

To evaluate how well the model makes right predictions compared to overall number of predictions, the accuracy metric is one of the most commonly used classification performance metrics. The evaluation is performed using a Confusion Matrix, which is presented in Table 3. This table visualizes the model's prediction results in comparison to the actual class labels [15]. Through a confusion matrix, accuracy measures the proportion of correct outputs relative to all predicted outputs. Although other metrics, accuracy provides an overview of the proportion of correct classifications among all tested data. The equation for accuracy is presented in the following calculation [15].

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

TABLE III
Confusion Matrix

Prediction	Actual	
	TRUE	**FALSE**
TRUE	True Positive (TP)	False Positive (FP)
FALSE	False Negative (FN)	True Negative (TN

K. Experiment Scenario Design

To obtain an optimal sentiment analysis model, this study designed five experimental scenarios that were conducted in stages. The first scenario was used to build a baseline model using three different architectures: CNN, LSTM, and CNN-LSTM hybrid, with k-fold cross validation to obtain a fair initial performance estimate. These three models used a dropout of 0.2 with sigmoid activation for the binary case, and were trained with 10 epochs and a batch size of 32. This research trained the models with hardware specifications of 12th Gen Intel(R) Core(TM) i5-12500H (16 CPUs) processor, 16,384MB RAM memory, and using NVIDIA GeForce RTX 3050. Model trained with K-fold Stratified Cross-Validation with k = 10. In each iteration, the model is trained on k-1 partitions of the data and evaluated on the remaining partition, repeating the cycle k times, so that each fold serves as a disposable test dataset. The final performance metric is an average of the results from all k iterations, thus providing a more reliable estimate and reducing the risk of bias that can occur with a single training-test split [36, 37]. The second scenario evaluates the representation of text features using various combinations of N-grams on TF-IDF vectorization technique, with the aim of finding the best configuration in capturing word context. In the third scenario, the maximum number of features (max features) in TF-IDF is set to avoid overfitting and maintain computational efficiency. The fourth scenario expands the feature dimension using Indonesian FastText embedding as a semantic expansion method to enrich the word representation. The last scenario, the fifth, focuses on optimizing the model training process by comparing five different optimization algorithms to find the most effective training approach in achieving optimal convergence and accuracy. These five scenarios are structured to ensure that performance improvements are made in a structured manner from the beginning to the end. A summary of the five scenarios is shown in Table 4.

TABLE IV
Experiment Summary

Scenario	Objective	Description
1	Establishing a baseline model for sentiment classification	Testing the initial performance of CNN, LSTM, and Hybrid CNN-LSTM with K-Fold CV
2	Optimizing TF-IDF feature representation with N-grams	Trying six combinations of N-grams to find the most accurate configuration
3	Determining the best number of features for TF-IDF	Applying max features variation to avoid overfitting
4	Expanding feature dimension with semantic representation	Using pre-trained Indonesian FastText as feature expansion
5	Maximizing model performance through optimization algorithm	Testing five types of optimizers to achieve the best convergence and accuracy

IV. Results And Discussion

In this research, five experimental scenarios were conducted to optimize model accuracy. First, baseline models were built using CNN, LSTM, and Hybrid CNN-LSTM with cross-validation to obtain fair initial performance estimates. In the second scenario, experiments were conducted on text feature representation with various combinations of N-grams. The third scenario continues the previous approach by adjusting the number of features retrieved by TF-IDF to remain informative. Furthermore, the fourth scenario enriches the feature dimensions using FastText as feature expansion method that captures deeper word meanings. Finally, the fifth scenario focuses on optimizing the training process through five types of optimization algorithm. This approach is designed to ensure that each component of the system is systematically improved to maximize performance.

A. Scenario 1

In the first scenario, sentiment classification experiments were conducted using three baseline architectures: CNN, LSTM, and Hybrid CNN-LSTM, which serves as a reference for accuracy improvement in the next scenario. The CNN architecture consists of Embedding, Conv1D, and GlobalMaxPooling1D. For the LSTM model, embedding layer initiates the model's architecture, followed by two sequential LSTM layers. Meanwhile, Hybrid CNN-LSTM combines the CNN in extracting local features with the ability of LSTM in capturing long-term dependencies.

TABLE V
FIRST SCENARIO RESULT

Fold	CNN		LSTM		CNN-LSTM	
	Accuracy (%)	**F1-Score (%)**	**Accuracy (%)**	**F1-Score (%)**	**Accuracy (%)**	**F1-Score (%)**
1	0.665	0.664	0.666	0.665	0.720	0.719
2	0.641	0.640	0.684	0.682	0.739	0.738
3	0.659	0.658	0.663	0.659	0.717	0.716
4	0.640	0.635	0.703	0.700	0.722	0.720
5	0.673	0.672	0.661	0.659	0.723	0.720
6	0.677	0.676	0.689	0.688	0.723	0.718
7	0.643	0.642	0.705	0.704	0.712	0.711
8	0.642	0.641	0.674	0.666	0.726	0.724
9	0.675	0.673	0.668	0.667	0.703	0.702
10	0.669	0.666	0.649	0.641	0.712	0.711
AVG	**0.6584**	**0.6574**	**0.6762**	**0.6737**	**0.7197**	**0.7185**
STD	0.01541427	0.015570	0.01854004	0.019875	0.00968446	0.009436

The results of evaluating the baseline architectures are shown in Table 5. The CNN model yielded the lowest average accuracy of 65.84%. The LSTM model showed slightly higher average accuracy, at 67.62%. The CNN-LSTM hybrid model produced the best performance, with an average accuracy of 71.97% with STD (Standar Deviation) of 0.0097 and F1-Score of 71.85% with 0.009436. This model demonstrated effective synergy between CNN local feature extraction and LSTM temporal modelling, achieving the highest consistency with the lowest standard deviation.

B. Scenario 2

In the second scenario, the three models are implemented with TF-IDF vectorization using different N-gram parameters to optimize the textual feature representation. This implementation includes six N-gram variations: unigram, bigram, trigram, as well as N-gram combinations. The model structure used is the same as in the first scenario. The results of the second scenario are presented in Table 6.

TABLE VI
SECOND SCENARIO RESULT

N-Gram	CNN			LSTM			CNN-LSTM		
	Accuracy	**STD**	**F1-Score**	**Accuracy**	**STD**	**F1-Score**	**Accuracy**	**STD**	**F1-Score**
(1,1)	0.7120 (+8.13)	0.0069	0.7095	0.7494 (+10.8)	0.0157	0.7489	0.7375 (+2.47)	0.0072	0.7367
(2,2)	0.6527 (-0.87)	0.0166	0.6389	0.6764 (+0.03)	0.0112	0.6697	0.6730 (-6.49)	0.0074	0.6629

(3,3)	0.5749 (-12.6)	0.0136	0.5133	0.5832 (-13.7)	0.0121	0.5240	0.7357 (+2.22)	0.0050	0.7342
(1,2)	**0.7121 (+8.15)**	0.0056	0.7065	**0.7514 (+11.1)**	0.0145	0.7512	**0.7421 (+3.11)**	0.0069	0.7408
(2,3)	0.6486 (-1.49)	0.0118	0.6370	0.6763 (+0.01)	0.0126	0.6686	0.6694 (-6.99)	0.0051	0.6580
(1,3)	0.7040 (+6.91)	0.0155	0.6994	0.7488 (+10.7)	0.0150	0.7485	0.7357 (+2.22)	0.0050	0.7342

The experimental results in the second scenario showed a consistent pattern that the unibigram combination has produced optimal performance for all model architectures with LSTM achieving the highest accuracy of 75.14% (increased 11.1% from baseline), followed by CNN-LSTM 74.21% (increased 3.11%), and CNN 71.21% (increased 8.15%). This showed that unigram features remain the basic representation containing individual word sentiment information, while bigram features provide enough local context to capture simple patterns.

C. Scenario 3

The third scenario is a continuation of the second scenario, which utilized the N-grams (1,2) because produced the highest accuracy in the previous experiment. In this scenario, the max features parameter was experimented with four variations: 1000, 2000, 5000, and 10000 features.

TABLE VII
Third Scenario Result

Max Features	CNN			LSTM			CNN-LSTM		
	Accuracy	**STD**	**F1-Score**	**Accuracy**	**STD**	**F1-Score**	**Accuracy**	**STD**	**F1-Score**
1000	0.7678 (+16.6)	0.0046	0.7666	0.7812 (+15.5)	0.0149	0.7804	0.7721 (+7.27)	0.0092	0.7711
2000	0.7711 (+17.1)	0.0112	0.7695	0.7815 (+15.5)	0.0091	0.7809	0.7774 (+8.01)	0.0076	0.7771
5000	0.7826 (+18.8)	0.0176	0.7815	**0.7854 (+16.1)**	0.0147	0.7839	0.7736 (+7.49)	0.0017	0.7730
10000	**0.7846 (+19.2)**	0.0072	0.7839	0.7822 (+15.6)	0.0121	0.7812	**0.7776 (+8.04)**	0.0057	0.7761

Table 7 showed different patterns between architectures in response to increasing TF-IDF feature dimensions. The CNN model demonstrated a consistent and monotonic performance improvement with the increase in max features, achieving an optimal accuracy of 78.46% (STD = 0.0072) at 10,000 features. In contrast, the LSTM model showed an optimal pattern at 5000 features with accuracy of 78.54%, then experienced a slight degradation at 10000 features (78.22%). On the other hand, the CNN-LSTM hybrid model did not consistently show superiority over the individual models, with accuracy of 77.76% increasing only 8.06% from the baseline.

D. Scenario 4

This scenario, FastText performed as a feature expansion technique to improve the semantic representation in sentiment classification. The implemented FastText model is pre-trained in Indonesian with an embedding dimension of 300. The model used follows the optimal configuration from the previous scenario. Table 8 shows the results of the fourth scenario experiment.

TABLE VIII
FOURTH SCENARIO RESULT

Model	Accuracy	STD	F1-Score
CNN	0.8269 (+25.5)	0.005292	0.8263
LSTM	0.8505 (+25.7)	0.003507	0.8500
CNN-LSTM	0.8491 (+17.9)	0.000047	0.8483

The experimental results of the fourth scenario showed significant performance improvement across all models utilizing FastText embeddings. The LSTM model reached the best accuracy of 85.05% with an STD of 0.0035. LSTM demonstrated its capability in this scenario with such accuracy. The remarkable phenomenon of CNN-LSTM, characterized by a very low standard deviation, indicates perfect performance consistency across all folds (84.91%). The substantial performance improvement in all architectures over the baseline demonstrates the effectiveness of feature expansion, rather than relying solely on TF-IDF.

E. Scenario 5

The fifth scenario is the final optimization stage that aims to maximize model performance through experiments with five different optimizer algorithms: Adam, SGD, RMSprop, Adagrad, and Adamax. This experiment uses the model architecture that has been proven optimal in the previous scenario.

TABLE IX
FIFTH SCENARIO RESULT

Optimizer	CNN			LSTM			CNN-LSTM		
	Accuracy	STD	F1-Score	Accuracy	STD	F1-Score	Accuracy	STD	F1-Score
Adam	0.8367 (+27.0)	0.005919	0.8361	0.8516 (+25.9)	0.004395	0.8514	0.8469 (+17.6)	0.001556	0.8467
SGD	0.8329 (+26.5)	0.018263	0.8316	0.8459 (+25.1)	0.006714	0.8452	0.8492 (+17.9)	0.003017	0.8492
RMSprop	**0.8441 (+28.1)**	0.003121	0.8435	0.8522 (+26.0)	0.005221	0.8518	**0.8884 (+23.4)**	0.000754	0.8880
Adagrad	0.8348 (+26.8)	0.003312	0.8339	0.8032 (+18.8)	0.133773	0.7791	0.8524 (+18.4)	0.000377	0.8521
Adamax	0.8390 (+27.4)	0.009352	0.8382	**0.8524 (+26.0)**	0.004232	0.8520	0.8509 (+18.2)	0.000471	0.8508

The experimental results of the fifth scenario on Table 9 show that the CNN-LSTM achieved a breakthrough performance of 88.84% using the RMSprop with an STD of 0.000754, which represents a significant improvement (23.4% from the baseline) and shows the lowest standard deviation. The CNN model demonstrated optimal performance with RMSprop with accuracy of 84.41%, while the LSTM achieved the highest accuracy with Adamax (accuracy 85.24%). The significant increase in accuracy was primarily attributed to RMSprop in architectures featuring convolutional layers, whereas LSTM showed a preference for Adamax.

V. CONCLUSION

This research developed a sentiment analysis system for the Mobile Legends: Bang Bang (MLBB) application by implementing a hybrid CNN-LSTM model with a systematic, stepwise optimization approach. The dataset used is from the Google Play Store, which was extracted through web scraping. The final pre-processed and balanced sampling dataset generates 30,000 instances with balanced distribution (15,000 positive and 15,000 negative), ensuring optimal representation for training binary classification models. Through five comprehensive experimental scenarios, this research demonstrates a significant performance improvement from the baseline CNN-LSTM model, achieving 71.97% accuracy, to a performance breakthrough of 88.84% using the RMSprop optimizer with a standard deviation of 0.000754, representing a 23.4% improvement in accuracy over the baseline model. Empirical validation demonstrated that the strategic combination of feature engineering through TF-IDF with N-grams (1, 2) and a maximum of 10,000 features, feature expansion using Indonesian FastText embeddings, and optimizer optimization with RMSprop yielded a good semantic representation and optimal convergence. The primary novelty of this research resides in systematic optimization methodology, which demonstrates that improvement approaches through feature engineering and algorithmic optimization can yield significant performance gains in sentiment analysis tasks. Although the method achieves 88.84% accuracy, it has challenges in handling data with sarcasm that contains sentiments that contradict the literal meaning, as well as mixed sentiment, where the model struggles to handle reviews that contain both positive and negative aspects. For future research, exploring advanced embedding techniques, such as BERT-based models, and extending them to multi-class sentiment classification with more detailed emotion granularity may offer deeper insights into intricate sentiment patterns.

Acknowledgment

The authors would like to express their deepest gratitude to all institutions and parties that have supported the completion of this research. Special appreciation goes to the academic and administrative community at Telkom University for their continued support. The authors are also grateful to the reviewers and peers for their constructive feedback, which has contributed significantly to the improvement of this work.

References

[1] V. H. Pranatawijaya, N. N. K. Sari, R. A. Rahman, E. Christian, and S. Geges, "Unveiling user sentiment: Aspect-Based analysis and topic modeling of Ride-Hailing and Google Play app reviews," *Journal of Information Systems Engineering and Business Intelligence*, vol. 10, no. 3, pp. 328–339, Oct. 2024, doi: 10.20473/jisebi.10.3.328-339.

[2] P. H. C. Samanmali and R. a. H. M. Rupasingha, "Sentiment analysis on google play store app users' reviews based on deep learning approach," *Multimedia Tools and Applications*, vol. 83, no. 36, pp. 84425–84453, Apr. 2024, doi: 10.1007/s11042-024-19185-w.

[3] A. S. Chan, F. Fachrizal, and A. R. Lubis, "Outcome prediction using Naïve Bayes algorithm in the selection of role Hero Mobile Legend," *Journal of Physics Conference Series*, vol. 1566, no. 1, p. 012041, Jun. 2020, doi: 10.1088/1742-6596/1566/1/012041.

[4] Y. Chandra and A. Jana, "Sentiment Analysis using Machine Learning and Deep Learning," *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2020, doi: 10.23919/indiacom49435.2020.9083703.

[5] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4335–4385, Dec. 2019, doi: 10.1007/s10462-019-09794-5.

[6] M. Sivakumar and S. R. Uyyala, "Aspect-based sentiment analysis of mobile phone reviews using LSTM and fuzzy logic," *International Journal of Data Science and Analytics*, vol. 12, no. 4, pp. 355–367, 2021, doi: 10.1007/s41060-021-00277-x.

[7] S. A. Thampy and J. J. R. Angelina, "Deep Learning Architectures Based Sentiment Analysis Systematic Literature Review*," 2023 International Conference on Control, Communication and Computing (ICCC)*, May 2023, doi: 10.1109/iccc57789.2023.10164943.

[8] N. M. A. Raihan and N. E. B. Setiawan, "Aspect Based Sentiment Analysis with FastText Feature Expansion and Support Vector Machine Method on Twitter," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 6, no. 4, pp. 591–598, Aug. 2022, doi: 10.29207/resti. v6i4.4187.

[9] H. Mohaouchane, A. Mourhir and N. S. Nikolov, "Detecting Offensive Language on Arabic Social Media Using Deep Learning," *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 2019, pp. 466-471, doi: 10.1109/SNAMS.2019.8931839.

[10] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jan. 2014, doi: 10.3115/v1/p14-1062.

[11] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning--based text classification," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–40, Apr. 2021, doi: 10.1145/3439726.

[12] S. Santhanam, "Context based Text-generation using LSTM networks," *arXiv*, Apr. 30, 2020. https://arxiv.org/abs/2005.00048.

[13] A. H. Ombabi, W. Ouarda, and A. M. Alimi, "Improving Arabic sentiment analysis across context-aware attention deep model based on natural language processing," *Language Resources and Evaluation*, Apr. 2024, doi: 10.1007/s10579-024-09741-z.

[14] A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, "A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26597–26613, Jun. 2019, doi: 10.1007/s11042-019-07788-7.

[15] N. Hossain, Md. R. Bhuiyan, Z. N. Tumpa, and S. A. Hossain, "Sentiment Analysis of Restaurant Reviews using Combined CNN-LSTM," *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–5, Jul. 2020, doi: 10.1109/icccnt49239.2020.9225328.

[16] B. M. G. A. Awienoor, and E. Setiawan, "Movie Recommendation System Based on Tweets Using Switching Hybrid Filtering with Recurrent Neural Network," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 2, pp. 277–293, Feb. 2024, doi: 10.22266/ijies2024.0430.24.

[17] F. Alharbi, L. Ouarbya, and J. A. Ward, “Comparing sampling strategies for tackling imbalanced data in human activity recognition,” *Sensors*, vol. 22, no. 4, p. 1373, Feb. 2022, doi: 10.3390/s22041373.

[18] F. Rifaldy, Y. Sibaroni, and S. S. Prasetiyowati, “Effectiveness of Word2VEC and TF-IDF in sentiment classification on online investment platforms using support Vector Machine,” *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, vol. 10, no. 2, pp. 863–874, Mar. 2025, doi: 10.29100/jipi. v10i2.6055.

[19] A. I. Kadhim, “Term weighting for feature extraction on Twitter: A comparison between BM25 and TF-IDF,” *2019 International Conference on Advanced Science and Engineering (ICOASE)*, pp. 124–128, Apr. 2019, doi: 10.1109/icoase.2019.8723825.

[20] N. I. R. Illahi and E. B. Setiawan, “Sentiment Analysis on Social Media Using Fasttext Feature Expansion and Recurrent Neural Network (RNN) with Genetic Algorithm Optimization,” *International Journal on Information and Communication Technology (IJoICT)*, vol. 10, no. 1, pp. 78–89, Jun. 2024, doi: 10.21108/ijoict. v10i1.905.

[21] N. Nedjah, I. Santos, and L. De Macedo Mourelle, “Sentiment analysis using convolutional neural network via word embeddings,” *Evolutionary Intelligence*, vol. 15, no. 4, pp. 2295–2319, Apr. 2019, doi: 10.1007/s12065-019-00227-4.

[22] M. Z. F. Rizky, Y. Sibaroni, and S. S. Prasetiyowati, “Sentiment Analysis on TikTok App using Long Short-Term Memory (LSTM) with Stochastic Gradient Descent (SGD) Optimization,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 3, p. 1292, Jul. 2024, doi: 10.30865/mib. v8i3.7699.

[23] N. Uly, H. Hendry, and A. Iriani, “CNN-RNN hybrid model for diagnosis of COVID-19 on X-Ray imagery,” *Digital Zone Jurnal Teknologi Informasi Dan Komunikasi*, vol. 14, no. 1, pp. 57–67, May 2023, doi: 10.31849/digitalzone. v14i1.13668.

[24] A. Ishaq, S. Asghar, and S. A. Gillani, “Aspect-Based sentiment analysis using a hybridized approach based on CNN and GA,” *IEEE Access*, vol. 8, pp. 135499–135512, Jan. 2020, doi: 10.1109/access.2020.3011802.

[25] B. Liu, “Text sentiment analysis based on CBOW model and deep learning in big data environment,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 2, pp. 451–458, Oct. 2018, doi: 10.1007/s12652-018-1095-6.

[26] M. Ihsan, N. B. S. Negara, and N. S. Agustian, “LSTM (Long Short-Term Memory) for Sentiment COVID-19 vaccine classification on Twitter,” *Digital Zone Jurnal Teknologi Informasi Dan Komunikasi*, vol. 13, no. 1, pp. 79–89, May 2022, doi: 10.31849/digitalzone. v13i1.9950.

[27] N. Dr. G. S. N. Murthy, N. S. R. Allu, N. B. Andhavarapu, and N. M. B. M. Belusonti, “Text based Sentiment Analysis using LSTM,” *International Journal of Engineering Research*, vol. V9, no. 05, May 2020, doi: 10.17577/ijertv9is050290.

[28] C. Sahoo, M. Wankhade, and B. K. Singh, “Sentiment analysis using deep learning techniques: a comprehensive review,” *International Journal of Multimedia Information Retrieval*, vol. 12, no. 2, Nov. 2023, doi: 10.1007/s13735-023-00308-2.

[29] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, “Tree-Structured Regional CNN-LSTM Model for dimensional Sentiment Analysis,” *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 28, pp. 581–591, Dec. 2019, doi: 10.1109/taslp.2019.2959251.

[30] W. Li, L. Zhu, Y. Shi, K. Guo, and E. Cambria, “User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models,” *Applied Soft Computing*, vol. 94, p. 106435, Jun. 2020, doi: 10.1016/j.asoc.2020.106435.

[31] A. B. Syahputri and Y. Sibaroni, “Comparative Analysis of CNN and LSTM Performance for Hate Speech Detection on Twitter,” *2023 11th International Conference on Information and Communication Technology (ICoICT)*, pp. 190–195, Aug. 2023, doi: 10.1109/icoict58202.2023.10262656.

[32] A. Bekkar, B. Hssina, S. Douzi, and K. Douzi, “Air-pollution prediction in smart city, deep learning approach,” *Journal of Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00548-1.

[33] A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, “A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis,” *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26597–26613, Jun. 2019, doi: 10.1007/s11042-019-07788-7.

[34] S. H. Eom, “Developing sentimental analysis system based on various optimizer,” *The International Journal of Internet, Broadcasting and Communication*, vol. 13, no. 1, pp. 100–106, Feb. 2021, doi: 10.7236/ijibc.2021.13.1.100.

[35] N. Fatima, “Enhancing performance of a deep neural network: A comparative analysis of optimization Algorithms,” *ADCAIJ ADVANCES IN DISTRIBUTED COMPUTING AND ARTIFICIAL INTELLIGENCE JOURNAL*, vol. 9, no. 2, pp. 79–90, Jun. 2020, doi: 10.14201/adcaij2020927990.

[36] T.-T. Wong and P.-Y. Yeh, “Reliable Accuracy Estimates from k-Fold Cross Validation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1586–1594, Apr. 2019, doi: 10.1109/tkde.2019.2912815.

[37] O. Chamorro-Atalaya et al., “K-Fold Cross-Validation through Identification of the Opinion Classification Algorithm for the Satisfaction of University Students,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 19, no. 11, Aug. 2023, doi: 10.3991/ijoe. v19i11.39887.