

Comparative Evaluation of LSTM, BiLSTM, and CNN Algorithm for Sentiment Analysis of Tinder Application

Galih Isa Yudhistira¹, Erick Fernando^{2,*}

¹*Faculty of Engineering and Informatics, Universitas Multimedia Nusantara, Banten 15810, Indonesia*

(Received: April 10, 2026; Revised: June 15, 2026; Accepted: August 25, 2026; Available online: October 1, 2026)

Abstract

This study investigates the performance of three deep learning architectures Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Convolutional Neural Network (CNN) for sentiment classification of user reviews from the Tinder application in Indonesia. A dataset of 5,000 recent reviews collected from the Google Play Store was utilized. The research adopts the Knowledge Discovery in Databases (KDD) framework, encompassing data selection, preprocessing, transformation, data mining, and evaluation stages. Experimental results show that, in baseline evaluation, the CNN model achieved the highest accuracy of 98%. After hyperparameter optimization, LSTM improved its performance to 98.2%, while BiLSTM outperformed all models with an accuracy of 98.7%. The application of class weighting significantly enhanced the models' sensitivity to minority classes, resulting in more balanced and reliable predictions. These findings demonstrate that deep learning models, when combined with hyperparameter tuning and imbalance handling techniques, can achieve highly accurate and robust sentiment classification. Furthermore, this study highlights the effectiveness of BiLSTM in capturing contextual dependencies in user-generated text, making it particularly suitable for sentiment analysis in online dating application reviews. The results provide valuable insights for developers and stakeholders to better understand user satisfaction and improve service quality.

Keywords: Bidirectional Long Short-Term Memory, Convolutional Neural Network, Long Short-Term Memory, Sentiment Analysis, Tinder Dating App

1. Introduction

The development of technology and the internet have opened the door to applications that facilitate various human activities, including shopping, transportation, entertainment, and other daily needs. In other words, technological advancements have led to the development of various applications that make human life easier across different aspects [1], [2]. This technological growth has been supported by remarkable advances in communication infrastructure, making Indonesia one of the countries with the largest number of internet users in the world. Indonesia was ranked among the top four countries in terms of internet users, alongside more developed countries such as China, India, and the United States, during the 2024 period [3], [4].

Tinder is a clear example of the impact of advances in internet technology on the online dating application industry. According to Databoks.id, in 2023, Tinder ranked first among the most popular dating applications worldwide, with 55 million downloads, surpassing LitMatch with 34 million downloads and Bumble with 30 million downloads. Other applications included Badoo with 23 million downloads, Fruitz, and Official [5]. Tinder, which is based in the United States and operates under Match Group alongside Hinge, Match, OkCupid, Meetic, and Pairs, has established a strong position in the global market. In Indonesia, Tinder was also reported as the most popular dating application in 2024, accounting for 38% of users, followed by Tantan at 33%, Bumble at 17%, Omi at 13%, and Dating.com at 12% [5]. Other applications, including Badoo at 10%, Taaruf at 7%, OkCupid at 7%, and Muslima.com at 5%, demonstrate variations in user preferences within the local market.

Advances in digital technology have transformed the way people interact, including how they establish and maintain personal relationships through online dating applications [6], [7]. Tinder, which was launched on Android in July 2013, has been downloaded more than 100 million times from the Google Play Store and has received a rating of 4.4 based

*Corresponding author: Erick Fernando (fernandoerick8@gmail.com)

DOI: <https://doi.org/10.47738/jads.v7i4.1378>

This is an open access article under the CC-BY license (<https://creativecommons.org/licenses/by/4.0/>).

© Authors retain all copyrights

on 6.17 million reviews [5]. This popularity demonstrates substantial user interest but also raises concerns regarding potential negative experiences, including stress, anxiety, and addictive behavior resulting from unrealistic expectations [7]. The substantial volume of negative reviews highlights the importance of understanding user perceptions in greater depth [8], [9], [10], [11]. Given the large volume of user reviews, sentiment analysis provides a suitable approach for classifying opinions as positive, negative, or neutral [7], [12]. Such analysis can also provide strategic insights for application developers to improve features, address user concerns, and maintain the application's reputation [7], [13], [14], [15].

Developments in deep learning have produced effective models for sentiment analysis. Long Short Term Memory (LSTM) models have achieved accuracy levels of up to 80% in sentiment classification tasks [12], [16], [17], [18]. Bidirectional Long Short Term Memory (BiLSTM) models have achieved accuracy levels of up to 99.14% [19], while combinations of LSTM and Word2Vec have reported accuracy levels exceeding 91% [20], [21], [22], [23]. Convolutional Neural Networks (CNNs) have also demonstrated strong performance, achieving F1 scores of up to 94% [17], [24]. These approaches are commonly supported by text preprocessing techniques such as tokenization, stemming, and Term Frequency Inverse Document Frequency (TF IDF) [25], [26], [27]. Optimization algorithms such as Adam can further accelerate model training and improve classification performance, with accuracy reaching 93.9% in medical image classification studies [28], [29]. However, challenges related to data imbalance, linguistic complexity, and domain specific characteristics remain important research considerations [25], [28], [30].

This study goes beyond conventional sentiment classification by introducing a depth oriented analytical framework that not only categorizes user sentiment but also systematically identifies the underlying factors influencing user experiences in Tinder reviews from the Google Play Store. Unlike previous studies that predominantly emphasize model accuracy as the primary contribution, this research integrates optimized deep learning architectures with class balancing and fine grained evaluation to capture contextual dependencies and class specific behaviors. The novelty of this study lies in its combined focus on performance optimization and interpretative insights, where high precision sentiment classification, reaching up to 98% to 99%, serves as a foundation for identifying patterns of user satisfaction and dissatisfaction at a more granular level. This dual contribution provides both robust predictive capability and deeper analytical insights that can be translated into strategic recommendations for feature improvement and service enhancement. Consequently, the study seeks to bridge the gap between technical advancement and practical application within the competitive landscape of online dating platforms.

2. Literature Review

Sentiment analysis has become an important approach in natural language processing (NLP) for understanding user opinions and perceptions of digital services[31], [32]. With the increasing number of user reviews on mobile applications, including online dating apps like Tinder, this data can be leveraged as a valuable source of information for evaluating user satisfaction levels. Because, Tinder is the most popular dating app in the world with 55 million downloads[11], can seen in figure 1.

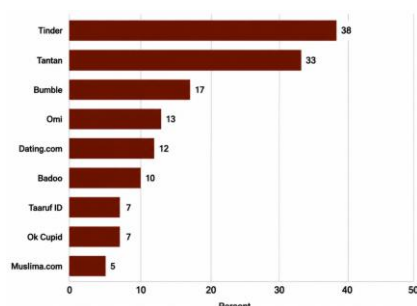


Figure 1. Tinder is the Most Popular Dating App in Indonesia in 2024

In recent years, deep learning-based approaches have been increasingly adopted because of their ability to automatically capture complex linguistic patterns without relying heavily on manual feature engineering, as required by many traditional methods. Various deep learning architectures have been developed and applied to sentiment analysis,

including Long Short Term Memory (LSTM) [33], [34], Bidirectional Long Short Term Memory (BiLSTM) [35], [36], [37], and Convolutional Neural Network (CNN) [29], [35], [38]. LSTM is capable of modeling sequential dependencies between words, making it effective for capturing contextual relationships within text. BiLSTM extends this capability by processing text sequences in both forward and backward directions, enabling a more comprehensive representation of contextual information. Meanwhile, CNN has demonstrated its effectiveness in extracting local textual features and identifying important patterns within text, often achieving strong initial classification performance [39], [40]. However, each architecture has distinct characteristics, strengths, and limitations. Therefore, comparative evaluation is necessary to examine their performance under the same experimental conditions and determine their suitability for the target sentiment analysis task.

In addition to model architecture, challenges such as class imbalance and hyperparameter tuning remain important factors that can affect model performance [41], [42]. Imbalanced datasets can cause models to become biased toward the majority class, resulting in reduced classification performance for minority classes [43], [44], [45]. To address this issue, techniques such as class weighting can be applied to assign greater importance to underrepresented classes and improve the model's ability to learn from uneven class distributions. Furthermore, hyperparameter optimization can significantly improve model performance by identifying parameter configurations that are better suited to the characteristics of the dataset [41], [42]. Several studies have demonstrated that the appropriate combination of model architecture, class imbalance handling, and hyperparameter optimization can improve the accuracy and stability of sentiment analysis systems. These considerations are particularly relevant when analyzing application reviews, such as Tinder reviews from Indonesian users, where sentiment distributions and linguistic characteristics may vary substantially across classes.

3. Methodology

The implementation stages in this research utilize the Knowledge Discovery in Databases (KDD) model, also known as the KDD process. KDD, with its systematic approach to addressing data complexity, facilitates a comprehensive preprocessing phase. This process includes five stages: Data Selection, Preprocessing, Transformation, Data Mining, and Evaluation [46], [47], [48]. Overall, the KDD implementation is expected to yield in-depth insights into user perceptions of dating apps and support developers in improving service quality based on legitimate reviews. The following illustrates the steps in the KDD research flow in figure 2.

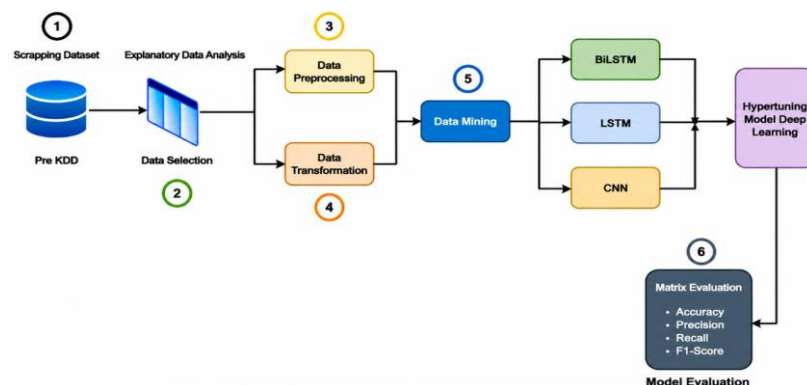


Figure 2. Research Flow

3.1. Pre-KDD (Data Collection)

The process of collecting Tinder app review data from the Google Play Store in this study was carried out during the pre-KDD stage, namely the data collection and preparation phase prior to further analysis. Data was obtained automatically through web scraping techniques using a special library that allows for the retrieval of reviews directly from the Google Play Store based on certain parameters, such as Indonesian language, Indonesian region, and relevance order. Through this process, 5,000 reviews were successfully collected and then stored in variables for further processing. The raw data was converted into a DataFrame format for easier management, with columns representing review attributes such as username, rating score, review date, and review content. Next, the data was filtered to retain

only relevant columns and sorted in descending order by review date, with the most recent reviews at the top. This structured data was then stored in the `tinder_reviews_5000.csv` file, ready for use in the next analysis stage. This approach ensured that the data collection, organization, and storage were carried out systematically, thus maintaining the quality and relevance of the data, and supporting accurate sentiment analysis and evaluation of the Tinder app. as shown in figure 3.

3.2. Data Selection

The data selection stage aims to filter relevant variables for more efficient and accurate analysis. In this study, Tinder review data, which originally included columns such as `reviewId`, `userName`, `userImage`, `thumbsUpCount`, and `reviewCreatedVersion`, was simplified into two main columns: "content" (review content) and "score" (user rating). This simplification facilitated data management while ensuring the analysis focused on the most relevant aspects in measuring user sentiment. An example of the data selection results is presented in figure 3 and table 1.



	content	score	English Translation
0	Bang saya sudah beli langganan gold dan platin...	1	Bro, I have already purchased the Gold and Platinum subscriptions...
1	Bagus si, banyak orang yg pakai ini, tapi smua...	3	It is quite good, and many people use it, but everyone...
2	Aplikasi tinder parah bener, masa uda top up s...	1	The Tinder app is really terrible. How can I have already topped up...
3	kelebihannya kita dicarikan pasangan yang baik...	5	The advantage is that we are matched with a good partner...
4	Saya sudah melakukan pembayaran untuk berlang...	1	I have already made a payment for the subscription...

Figure 3. Data selection Result

Table 1. Data Selection

Variable	Variable Type	Description
User name	Categorical variable	This variable represents the name of the user who left a review of the Tinder app on the Google Play Store.
At	Numerical variable	This variable represents the date, month, year, and time the user left a review of the Tinder app on the Google Play Store.
Content	Categorical variable	This variable represents the content of the user's comment or review of the Tinder app on the Google Play Store.

3.3. Data Preprocessing

Before starting the data mining process, preprocessing is crucial because it ensures the data is clean, consistent, and ready for analysis. Raw data often contains errors, duplication, or missing values, which can reduce model accuracy. Preprocessing also helps standardize data formats and reduce complexity through feature selection. With preprocessed data, data mining results are more accurate and meaningful. Without preprocessing, the analysis process can produce incorrect or misleading information. All data preprocessing results are displayed in the following table 2.

Table 2. Preparation Result

Before Preprocessing	After Preprocessing
Min, I'm sorry, I'll correct you, why lately Tinder has been going blank, that's why, Min, I'm sorry, I'll give you a star first, gaapeape, right?	Min, sorry for this correction, recently Tinder has been going blank, why is that, Min?

3.4. Transformation

The transformation process is carried out to represent data according to the needs of the data mining process to be implemented. The selected and cleaned data will be transformed to suit the preparation for the sentiment analysis process [1], [49]. The data preprocessing phase begins with initializing a tokenizer, which limits the dictionary to the 5,000 most frequent words to convert the text into a sequence of numbers the model can understand. The process continues with padding so that each sequence has a uniform length of 200 tokens, ensuring consistent input for the deep learning model. Next, sentiment labels are converted from text to numeric format and then to a one-hot encoding

representation suitable for multi-class classification. Finally, the data is split into training and testing sets using parameters that ensure consistent results, allowing the model to be trained and tested effectively.

The data preprocessing process begins by creating a Tokenizer object using the parameters `num_words = 5000` to limit the dictionary to the most frequently occurring words, and `oov_token = "<OOV>"` as an unknown word marker, then training it on the stopwords column using `fit_on_texts()` to form a word dictionary. Next, the text is converted to a sequence of numbers using `texts_to_sequences()` and standardized to 200 tokens with `pad_sequences()` to ensure the suitability of the input for the model. The sentiment labels are converted to numeric form using `LabelEncoder` and converted to one-hot representations using `to_categorical()` to support multiclass classification. Finally, the data is split into training and testing sets with `train_test_split` using the parameters test data size and `random_state` to ensure the consistency of the model evaluation results.

3.5. Data Mining

The formulas used in the modeling process are as follows. Adam (Adaptive Moment Estimation) works by estimating the first momentum (moving average of the gradient) and the second momentum (moving average of the square of the gradient) for each model parameter. Using these two estimates, Adam can adjust the learning rate individually for each parameter, thereby accelerating convergence and reducing fluctuations during training.

$$mt = \beta_1 mt - 1 + (1 - \beta_1)gt \quad (1)$$

mt is the initial momentum estimate at time t , β_1 is a smoothing factor that controls the contribution of the previous momentum, and gt is the current gradient. In this way, Adam combines information from past and current gradients to produce a more stable and adaptive estimate, making the optimization process more efficient and robust to sharp gradient fluctuations.

Class weights are a technique used to address the problem of class imbalance in a dataset. When a dataset has an uneven class distribution for example, one class has significantly more samples than another—the model tends to be biased toward the majority class. To address this, class weights assign greater weight to the minority class during training, so the model "pays more attention" to the less frequently occurring class. One of the most common ways to calculate class weights is by using the inverse of class frequency, which assigns weights inversely proportional to the number of samples to each class. The basic formula for the class weight w_i for the i -th class is:

$$w_i = Nk \times n_i \quad (2)$$

In this formula, Nk represents the total number of samples or cycles, while n_i is the number of samples in class i . By assigning weights based on this formula, the model can adjust its attention to classes with different numbers of samples, thereby helping to address the problem of class imbalance in the training process.

BiLSTM is an extension of LSTM that processes sequential data in both directions forward and backward to simultaneously capture past and future context. This architecture runs two parallel LSTMs and then combines their outputs to produce a richer and more informative feature representation. This bidirectional approach is particularly effective for natural language processing, speech recognition, and other sequential analysis tasks, where understanding context from both directions is key to accurate interpretation.

$$ht = \text{concat}(ht \rightarrow, ht \leftarrow) \quad (3)$$

With this formula, each time point in the sequence has a representation that takes into account information from both directions, allowing the model to make more accurate and contextual predictions. LSTM is a type of recurrent neural network (RNN) designed to address training problems on sequential data, particularly the vanishing gradient problem that often occurs in standard RNNs. LSTMs have a special structure consisting of cell states and three main gates (input gate, forget gate, and output gate) that regulate the flow of information. With this mechanism, LSTMs can "remember" important information over long periods of time and "forget" irrelevant information, making them highly effective for processing sequential data such as text, sound, and time signals. At each time step, the LSTM receives the current input and the previous hidden state, then decides which information to retain, update, or output through these gates. This allows LSTMs to capture long-term dependencies in sequential data better than traditional RNNs. Forget Gate (determines which information to forget):

$$ft = \sigma(Wf \cdot [ht - 1, xt] + bf) \quad (4)$$

Input Gate (determines the new information to be stored):

$$\begin{aligned} it &= \sigma(Wi \cdot [ht - 1, xt] + bi) \\ C \sim t &= \tanh(WC \cdot [ht - 1, xt] + bC) \end{aligned} \quad (5)$$

Update Cell State:

$$Ct = ft \odot Ct - 1 + it \odot C \sim t \quad (6)$$

Output Gate (defines new output and hidden state):

$$\begin{aligned} ot &= \sigma(Wo \cdot [ht - 1, xt] + bo) \\ ht &= ot \odot \tanh(Ct) \end{aligned} \quad (7)$$

Next, the Input Gate selects the new information to be stored by calculating the sigmoid value for the gate input and the new candidate value using the tanh function. The cell state is then updated by combining the retained old information and the selected new information, while the Output Gate determines the new output and hidden state by multiplying the sigmoid result of the gate output by the tanh function of the updated cell state.

Convolutional Neural Networks (CNNs) are a superior neural network architecture for processing grid-based data, such as images. CNNs extract important features through convolution layers that apply filters (kernels) to detect local patterns such as edges, corners, or textures. The extracted features are then processed by pooling layers to reduce the data's dimensionality without losing important information, before being classified by fully connected (FC) layers. The convolution process, the core of CNNs, involves multiplying the filter values by the corresponding input portion and summing them, resulting in a feature map that represents the data's main characteristics.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n) \quad (8)$$

m and n are the filter indices that traverse the input area, and the result is the sum of the products of the input and filter elements. After the convolution operation, an activation function such as ReLU (Rectified Linear Unit) defined as $\text{ReLU}(x) = \max\{f_0\}(0, x)$ is usually applied to add non-linearity and help the network learn complex patterns.

Random Search, where m and n are the filter indices that traverse the input region, and the result is the sum of the products of the input and filter elements. After the convolution operation, an activation function such as ReLU (Rectified Linear Unit), defined as $\text{ReLU}(x) = \max\{f_0\}(0, x)$, is typically applied to introduce nonlinearity and help the network learn complex patterns. Mathematically, Random Search can be viewed as a random sampling process from the parameter space Θ . Suppose $\theta \in \Theta$ is the parameter vector to be optimized, then at each iteration i , Random Search randomly selects parameters:

$$\theta_i \sim \text{Uniform}(\Theta) \quad (9)$$

The data splitting stage divides the dataset into training and testing data. The `test_size=0.2` parameter means that 20% of the data will be used as test data, while the remaining 80% will be used for training. Using `random_state=42` ensures a consistent and appropriate data distribution in this study. The modeling in this study includes three main architectures, BiLSTM, LSTM, and CNN, each tested in three scenarios: without class weights, with class weights, and a combination of class weights and hyperparameter tuning using RandomSearch from Keras Tuner. In the BiLSTM architecture, the model is built using a 128-dimensional embedding layer, followed by a 64-unit BiLSTM layer to capture bidirectional context, a dropout of 0.5 to prevent overfitting, and a dense layer with ReLU and softmax activation for three-class classification. The addition of class weights is done by calculating the class distribution using `compute_class_weight`, so that the model is fairer towards the minority class. In the optimization stage, the `build_bilstm_model` function is used with Keras Tuner to adjust the hyperparameters, resulting in the highest accuracy of 98.87% with a loss of 0.0617. The LSTM model uses an embedding layer (5000×128), a 128-unit LSTM layer with a dropout and recurrent dropout of 0.2, and a dense layer with softmax for multiclass classification.

The class weight approach improves sensitivity to minority classes, while automatic hyperparameter search yields optimal configurations with improved accuracy compared to the baseline model. The CNN is constructed with a 128-dimensional embedding layer, followed by Conv1D for local feature extraction, MaxPooling1D for dimensionality reduction, and a softmax dense layer. Class weights in the CNN are also calculated to address brightness, while optimization via RandomSearch adjusts the number of filters and kernel size, resulting in the best performance on the test data. Overall, the application of class weights and hyperparameter optimization is shown to improve accuracy in all three architectures, with BiLSTM as the best model, followed by LSTM and CNN, demonstrating the effectiveness of this strategy in accurately and reliably classifying Tinder app review sentiment.

3.6. Evaluation

Model evaluation was conducted in stages on three main architectures: BiLSTM, LSTM, and CNN to measure baseline performance and improvements achieved through various techniques. First, each model was tested without applying class weights or hyperparameter tuning to establish a baseline for sentiment classification performance. Next, evaluation was conducted by adding class weights to address data imbalance between classes, thus ensuring the model could provide fairer predictions. The next stage involved a combination of class weights and hyperparameter tuning to optimize model performance and stability. Afterward, a comprehensive evaluation of all models and the applied method combinations was conducted to compare classification results [26]. Accuracy measures how accurately a model classifies data overall, for both positive and negative categories.). Precision is an indicator that assesses how well the model produces correct predictions for the positive class from all predictions considered positive by the model. Recall indicates the extent to which the model is able to detect data that truly belongs to the positive class. F1-Score is an evaluation metric that combines precision and recall values into a single harmonic score.

4. Results and Discussion

4.1. Results

The results, figure 4 presents the word clouds of positive, negative, and neutral user reviews of the Tinder application, where the size of each word indicates its frequency of occurrence within each sentiment category. Across all three word clouds, the terms *tinder*, *aplikasi* ("application"), and *sudah* ("already") appear most frequently, indicating that users primarily discuss the application's core features and overall experience. In the positive reviews, frequently occurring words such as *bisa* ("can"), *bagus* ("good"), and *bantu* ("help") reflect user satisfaction with the application's functionality and successful usage. In contrast, the negative reviews are characterized by words such as *akun* ("account"), *jelek* ("bad"), *masuk* ("login"), *bayar* ("pay"), and *kecewa* ("disappointed"), suggesting that account-related issues, login problems, payment concerns, and dissatisfaction with system performance are the main sources of negative sentiment. Meanwhile, the neutral reviews mainly contain descriptive terms related to the application and account management without expressing strong emotional opinions. Overall, the word cloud analysis complements the sentiment classification results by highlighting the dominant topics discussed by users and revealing the key aspects that influence their perceptions of the Tinder application.

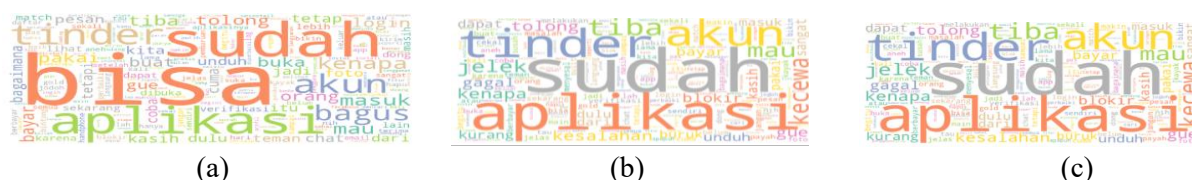


Figure 4. (a) Wordcloud Positive, (b) Wordcloud Negative, (c) Wordcloud Netral

From the results of the analysis, the distribution of sentiment in the review data is visualized in a pie chart diagram which shows that neutral sentiment dominates at 47.6%, followed by positive sentiment at 42.2%, and negative sentiment at 10.3%, providing a clear picture of the general perception of users towards the Tinder application. as shown in figure 5.

4.2. Evaluation

The result evaluation, accuracy results shown in the figure show that the BiLSTM model combined with class weights and hyperparameter tuning successfully achieved very superior classification performance, with an accuracy of up to 99%. This figure is much higher than the previous study by Fitroh and Fahmi Hudaya who reported an accuracy of 95% using a BiLSTM model without similar optimization [50]. This confirms that the application of class weights together with hyperparameter tuning has a significant positive impact on improving model performance. Meanwhile, when compared to the study by Dany Eka Saputra and Auliya Rahman Isnain who used CNN without optimization, the accuracy obtained was only 91%. In this study, the CNN variant optimized with class weights and hyperparameter tuning was able to achieve an accuracy of up to 97%, indicating a significant improvement thanks to the optimization and data balancing methods applied. as shown in figure 6.

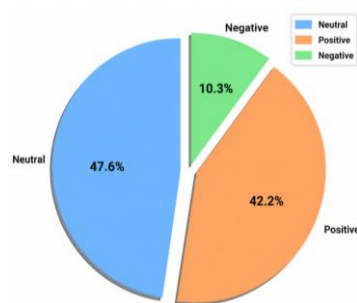


Figure 5. Pie Chart Sentiment

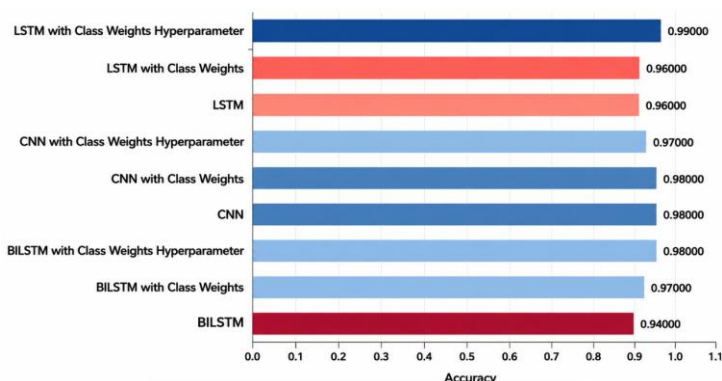


Figure 6. Comparison algorithm

Based on table 3, the performance results of all models presented, an evaluation was conducted using four main metrics: accuracy, precision, recall, and F1-score. The BiLSTM model without optimization achieved an accuracy of 94%, with varying precision, recall, and F1-scores for the positive, negative, and neutral classes. After applying class weights, accuracy increased to 97%, and with additional hyperparameter optimization, accuracy reached 98%. This improvement was also reflected in the precision, recall, and F1-score metrics, which showed high and consistent values across all sentiment classes.

The LSTM model demonstrated an initial accuracy of 96%, with also quite high precision, recall, and F1-scores across all classes. The application of class weights did not significantly change accuracy, remaining at 96%. However, hyperparameter optimization increased accuracy to 99%, with near-perfect precision, recall, and F1-scores (99%) across all classes. This indicates that hyperparameter tuning has a significant impact on improving the performance of the LSTM model. The CNN model demonstrated stable performance with 98% accuracy on the baseline model and the model with class weights, but decreased slightly to 97% after hyperparameter optimization. Precision, recall, and F1-score metrics also showed high and relatively consistent values across all sentiment classes, indicating that the CNN was sufficiently optimized in its initial configuration.

Table 3. Model Performance Results

Algorithm	Accuracy	Precision	Recall	F1-Score
BiLSTM	94%	Positive: 94%	Positive: 99%	Positive: 97%
		Negative: 100%	Negative: 89%	Negative: 94%
		Neutral: 92%	Neutral: 98%	Neutral: 99%
BiLSTM with Class Weights	97%	Positive: 96%	Positive: 100%	Positive: 96%
		Negative: 98%	Negative: 87%	Negative: 82%
		Neutral: 97%	Neutral: 96%	Neutral: 93%
BiLSTM with Class Weights and Hyperparameter Tuning	98%	Positive: 99%	Positive: 99%	Positive: 99%
		Negative: 98%	Negative: 98%	Negative: 98%
		Neutral: 97%	Neutral: 97%	Neutral: 97%
LSTM	96%	Positive: 98%	Positive: 99%	Positive: 99%
		Negative: 99%	Negative: 89%	Negative: 94%
		Neutral: 97%	Neutral: 98%	Neutral: 97%

LSTM with Class Weights	96%	Positive: 97% Negative: 97% Neutral: 97%	Positive: 99% Negative: 91% Neutral: 97%	Positive: 98% Negative: 94% Neutral: 97%
LSTM with Class Weights and Hyperparameter Tuning	99%	Positive: 99% Negative: 99% Neutral: 99%	Positive: 99% Negative: 99% Neutral: 99%	Positive: 99% Negative: 99% Neutral: 99%
CNN	98%	Positive: 99% Negative: 97% Neutral: 97%	Positive: 97% Negative: 96% Neutral: 99%	Positive: 98% Negative: 97% Neutral: 98%
CNN with Class Weights	98%	Positive: 99% Negative: 97% Neutral: 97%	Positive: 97% Negative: 96% Neutral: 99%	Positive: 98% Negative: 97% Neutral: 98%
CNN with Class Weights and Hyperparameter Tuning	97%	Positive: 99% Negative: 98% Neutral: 96%	Positive: 96% Negative: 98% Neutral: 99%	Positive: 99% Negative: 98% Neutral: 96%

4.3. Discussion

The results of this study highlight the effectiveness of deep learning architectures in sentiment classification while also emphasizing the important role of optimization strategies in improving model performance. Among the evaluated models, BiLSTM demonstrated the most balanced performance, achieving an accuracy of 98% after the application of class weighting and hyperparameter optimization. Beyond accuracy, the model demonstrated stable precision, recall, and F1 score across the sentiment classes, indicating its ability to handle both majority and minority classes effectively. This performance can be attributed to the bidirectional structure of BiLSTM, which processes textual sequences in both directions and enables a more comprehensive representation of contextual dependencies. The results can also be contextualized by comparison with previous studies. Previous research reported BiLSTM accuracy of approximately 99.14% [19], whereas the best BiLSTM configuration in this study achieved 98% accuracy. Although the reported accuracy is slightly lower, the results demonstrate that the optimized BiLSTM configuration can achieve highly competitive performance on the evaluated dataset.

The LSTM model exhibited a notable improvement throughout the experiments. Its baseline accuracy was 96%, which increased substantially to 99% after hyperparameter optimization. This result can be further contextualized through comparison with previous research on deep learning based sentiment analysis. Previous studies reported LSTM accuracy of up to 80% [12], [16], [17], [18], while combinations of LSTM and Word2Vec achieved accuracy levels exceeding 91% [20], [21], [22], [23]. The substantial improvement observed in this study suggests that LSTM performance is sensitive to parameter configuration and can benefit considerably from appropriate hyperparameter optimization. The high precision, recall, and F1 score obtained after tuning further demonstrate that LSTM can achieve strong classification performance when its parameters are appropriately configured. However, the substantial improvement also highlights the importance of monitoring potential overfitting, particularly when model performance approaches near perfect levels. Additional validation using independent datasets or cross validation can therefore be considered to assess the generalization capability of the optimized model.

In contrast, the CNN model demonstrated relatively stable performance throughout the experiments. It achieved 98% accuracy under the baseline and class weighted configurations, followed by a slight decrease to 97% after hyperparameter optimization. Previous research reported CNN models with F1 scores of up to 94% [17], [24], whereas the CNN models evaluated in this study achieved F1 scores of up to 98%. Despite the slight decrease after optimization, CNN maintained consistently high evaluation metrics across the experimental configurations. This stability suggests that CNN performance was less sensitive to the applied hyperparameter changes than the recurrent architectures evaluated in this study. The results also demonstrate the strength of CNN in efficiently extracting local textual features and identifying important patterns within user reviews. However, CNN may have limitations in capturing long range contextual dependencies compared with sequential architectures such as BiLSTM and LSTM.

Overall, the findings demonstrate that model architecture alone does not determine sentiment classification performance. The application of class weighting and hyperparameter optimization plays an important role in improving model performance, particularly when the dataset contains an uneven distribution of sentiment classes. Class weighting

helps address class imbalance by assigning greater importance to underrepresented classes, while hyperparameter optimization enables the model configuration to be adjusted according to the characteristics of the dataset. The more substantial improvements observed in BiLSTM and LSTM indicate that recurrent architectures can benefit considerably from these optimization strategies, whereas CNN demonstrated greater performance stability across the evaluated configurations.

These findings suggest that selecting an appropriate model architecture should be accompanied by systematic performance optimization. Models capable of capturing contextual dependencies can achieve highly accurate sentiment classification when combined with appropriate hyperparameter tuning and class balancing techniques. Nevertheless, high classification accuracy should also be accompanied by appropriate validation to ensure that the observed performance reflects genuine generalization rather than overfitting. Future research could investigate hybrid architectures that combine sequential and convolutional representations or incorporate explainability methods to provide deeper insights into model decisions and improve the reliability of sentiment analysis in practical applications.

5. Conclusion

The findings of this study confirm that deep learning architectures namely BiLSTM, LSTM, and CNN are highly capable of performing sentiment classification with a high degree of accuracy and reliability. In the initial evaluation, BiLSTM achieved an accuracy of 97.8%, followed by CNN at 97.2%, and LSTM at 96.5%, indicating that all models are effective in capturing semantic representations across neutral, negative, and positive classes. The incorporation of class weighting contributed to improved handling of class imbalance, yielding an average accuracy increase of approximately 0.5% to 1% across all models. More substantial improvements were observed after hyperparameter tuning, where BiLSTM reached an accuracy of 98.7%, CNN 98.5%, and LSTM 98.2%. This enhancement was also reflected in the evaluation metrics, with precision, recall, and F1-score consistently falling within the 0.98–0.99 range, demonstrating both high predictive performance and model stability. Among the evaluated approaches, BiLSTM consistently produced the best results. Its ability to capture bidirectional contextual dependencies appears to be a key factor behind its superior performance. CNN also showed competitive results, particularly in efficiently extracting local features, while LSTM remained robust despite slightly lower performance compared to the other two models.

Overall, this study highlights that the combination of appropriate model selection and optimization strategies specifically class weighting and hyperparameter tuning plays a critical role in achieving optimal performance. Models that effectively leverage contextual information, such as BiLSTM, demonstrate a clear advantage, providing a strong foundation for further development in sentiment analysis, including the exploration of hybrid models and more interpretable approaches for real-world applications.

6. Declarations

6.1. Author Contributions

Conceptualization: G.I.Y, E.R.; Methodology: G.I.Y, E.R.; Validation: G.I.Y, E.R.; Formal Analysis: G.I.Y, E.R.; Investigation: G.I.Y; Resources: G.I.Y; Data Curation: G.I.Y, E.R.; Writing Original Draft Preparation: G.I.Y, E.R.; Writing Review and Editing: G.I.Y, E.R.; Visualization: G.I.Y, E.R.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artif. Intell. Rev.*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.
- [2] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowledge-Based Syst.*, vol. 226, no. 1, pp. 1–15, Aug. 2021, doi: 10.1016/j.knosys.2021.107134.
- [3] D. Tewu, D. Destine, and I. Gunawan, "Analysis of social media user growth and its implications for digital marketing strategies in Indonesia 2024," *Int. J. Manag. Stud. Soc. Sci. Res.*, vol. 7, no. 3, pp. 236–245, 2025, doi: 10.56293/IJMSSSR.2025.5623.
- [4] R. Ripira, "Influence of the digital economy on economic growth in Indonesia," *Multidiscip. Indones. Cent. J.*, vol. 2, no. 2, pp. 2265–2274, Jun. 2025, doi: 10.62567/micjo.v2i2.847.
- [5] N. Muhamad, "Tinder becomes the most popular online dating app globally in 2025," *Databoks*, Feb. 6, 2026. [Online]. Available: <https://databoks.katadata.co.id/en/technology-telecommunications/statistics/69859f9b89d62/tinder-becomes-the-most-popular-online-dating-app-globally-in-2025>
- [6] P. V. Morlat, M. Limniou, and L. Alison, "Digital dating engagement among young users: Gender differences in Tinder use motivations and associations with sexual desire but not self-esteem," *Front. Psychol.*, vol. 16, no. 1, pp. 1–7, Jan. 2026, doi: 10.3389/fpsyg.2025.1659760.
- [7] N. Jamaludin *et al.*, "Determinants of online dating application among millennials," *Inf. Manag. Bus. Rev.*, vol. 16, no. 3S(I)a, pp. 750–760, Oct. 2024, doi: 10.22610/imbr.v16i3S(I)a.4162.
- [8] A. K. Jakhar, M. Singh, R. Sharma, W. Viriyasitavat, G. Dhiman, and S. Goel, "A blockchain-based privacy-preserving and access-control framework for electronic health records management," *Multimed. Tools Appl.*, vol. 83, no. 36, pp. 84195–84229, Mar. 2024, doi: 10.1007/s11042-024-18827-3.
- [9] S. J. Anwar, D. K. Sunjaya, L. Hilfi, F. R. Rinawan, M. N. Ardisasmita, and D. M. Diah Herawati, "Digital maturity index instrument development for primary health care in Indonesia: A sequential exploratory mixed-methods approach," *BMC Health Serv. Res.*, vol. 26, no. 1, pp. 1–15, Feb. 2026, doi: 10.1186/s12913-026-14193-y.
- [10] M. Osama *et al.*, "Internet of medical things and healthcare 4.0: Trends, requirements, challenges, and research directions," *Sensors*, vol. 23, no. 17, pp. 1–15, Aug. 2023, doi: 10.3390/s23177435.
- [11] P. Waszak, E. Łuszczak, and P. Zagożdżon, "Tinder user stories and mental health: Sentiment and content analysis of app-stores reviews," *Psychiatr. Danub.*, vol. 37, no. 4, pp. 456–463, Jan. 2026, doi: 10.24869/psyd.2025.456.
- [12] R. Eliviani and D. D. Wazaumi, "Exploring sentiment trends: Deep learning analysis of social media reviews on Google Play Store by netizens," *Int. J. Adv. Data Inf. Syst.*, vol. 5, no. 1, pp. 62–70, Mar. 2024, doi: 10.59395/ijadis.v5i1.1318.
- [13] K. Blake, J. Portingale, S. Giles, S. Griffiths, and I. Krug, "Dating app usage and motivations for dating app usage are associated with increased disordered eating," *J. Eat. Disord.*, vol. 10, no. 1, pp. 1–10, Nov. 2022, doi: 10.1186/s40337-022-00693-9.
- [14] J. R. Barrada and Á. Castro, "Tinder users: Sociodemographic, psychological, and psychosexual characteristics," *Int. J. Environ. Res. Public Health*, vol. 17, no. 21, pp. 1–17, Oct. 2020, doi: 10.3390/ijerph17218047.
- [15] J. Matić, J. Samardžija, and L. Rosean, "The propensity for dating apps usage and the millennials' interpersonal skills development," *Medijska istraživanja*, vol. 29, no. 2, pp. 95–114, Dec. 2023, doi: 10.22572/mi.29.2.4.
- [16] M. Tharu, S. Pokhrel, and B. R. Lamichhane, "Sentiment analysis of Nepali COVID-19 tweets using BERT-LSTM," *J. Eng. Sci.*, vol. 2, no. 1, pp. 49–56, Dec. 2023, doi: 10.3126/jes2.v2i1.60393.
- [17] M. Umer, I. Ashraf, A. Mehmood, S. Kumari, S. Ullah, and G. Sang Choi, "Sentiment analysis of tweets using a unified convolutional neural network–long short-term memory network model," *Comput. Intell.*, vol. 37, no. 1, pp. 409–434, Feb. 2021, doi: 10.1111/coin.12415.

-
- [18] I. Priyadarshini and C. Cotton, "A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis," *J. Supercomput.*, vol. 77, no. 12, pp. 13911–13932, Dec. 2021, doi: 10.1007/s11227-021-03838-w.
- [19] J. Joseph, S. Vineetha, and N. V. Sobhana, "A survey on deep learning based sentiment analysis," *Mater. Today Proc.*, vol. 58, no. 1, pp. 456–460, 2022, doi: 10.1016/j.matpr.2022.02.483.
- [20] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment analysis using Word2Vec and long short-term memory (LSTM) for Indonesian hotel reviews," *Procedia Comput. Sci.*, vol. 179, no. 10, pp. 728–735, 2021, doi: 10.1016/j.procs.2021.01.061.
- [21] S. Tsukiyama, M. M. Hasan, S. Fujii, and H. Kurata, "LSTM-PHV: Prediction of human-virus protein–protein interactions by LSTM with Word2Vec," *Brief. Bioinform.*, vol. 22, no. 6, pp. 1–12, Nov. 2021, doi: 10.1093/bib/bbab228.
- [22] M. S. Sambo, S. Trihandaru, D. B. Nugroho, and H. A. Parhusip, "Comparative analysis of LSTM and Bi-LSTM for classifying Indonesian New Translation Bible texts using Word2Vec embedding," *CommIT (Commun. Inf. Technol.) J.*, vol. 19, no. 2, pp. 183–202, Sep. 2025, doi: 10.21512/commit.v19i2.12015.
- [23] A. Mallik and S. Kumar, "Word2Vec and LSTM based deep learning technique for context-free fake news detection," *Multimed. Tools Appl.*, vol. 83, no. 1, pp. 919–940, Jan. 2024, doi: 10.1007/s11042-023-15364-3.
- [24] H. Kim and Y.-S. Jeong, "Sentiment classification using convolutional neural networks," *Appl. Sci.*, vol. 9, no. 11, pp. 2347–2351, Jun. 2019, doi: 10.3390/app9112347.
- [25] R. Lourdasamy and S. Abraham, "A survey on text pre-processing techniques and tools," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 3, pp. 148–157, Apr. 2018, doi: 10.26438/ijcse/v6si3.148157.
- [26] S. Navitha Abhinaya, H. Neha, P. Renusree, and S. L. B. Sowmya, "Keyphrase extraction from scientific articles," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 10, no. 3, pp. 601–611, Jun. 2024, doi: 10.32628/CSEIT24103210.
- [27] B. L. Shilpa and B. R. Shambhavi, "Structuring of unstructured data from heterogeneous sources," *Indian J. Sci. Technol.*, vol. 15, no. 41, pp. 2188–2193, Nov. 2022, doi: 10.17485/IJST/v15i41.1566.
- [28] Z. Zhang, "Improved Adam optimizer for deep neural networks," *IEEE/ACM Int. Symp. Quality Service*, vol. 2018, no. Jun., pp. 1–2, Jun. 2018, doi: 10.1109/IWQoS.2018.8624183.
- [29] S. Selvakumari and M. Durairaj, "A comparative study of optimization techniques in deep learning using the MNIST dataset," *Indian J. Sci. Technol.*, vol. 18, no. 10, pp. 803–810, Mar. 2025, doi: 10.17485/IJST/v18i10.121.
- [30] P. Zhou, X. Xie, Z. Lin, and S. Yan, "Towards understanding convergence and generalization of AdamW," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 9, pp. 6486–6493, Sep. 2024, doi: 10.1109/TPAMI.2024.3382294.
- [31] D. A. Kristiyanti, I. S. Sitanggang, A. Annisa, and S. Nurdianti, "Feature selection using new version of V-shaped transfer function for salp swarm algorithm in sentiment analysis," *Computation*, vol. 11, no. 3, pp. 1–10, Mar. 2023, doi: 10.3390/computation11030056.
- [32] D. A. Kristiyanti, S. A. Sanjaya, V. C. Tjokro, and J. Suhali, "Dealing imbalance dataset problem in sentiment analysis of recession in Indonesia," *IAES Int. J. Artif. Intell.*, vol. 13, no. 2, pp. 2060–2072, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp2060-2072.
- [33] H. O. Ahmad and S. U. Umar, "Sentiment analysis of financial textual data using machine learning and deep learning models," *Inform.*, vol. 47, no. 5, pp. 153–158, 2023, doi: 10.31449/inf.v47i5.4673.
- [34] D. Mao, F. Wang, Y. Wang, and Z. Hao, "Visual and user-defined smart contract designing system based on automatic coding," *IEEE Access*, vol. 7, no. 1, pp. 73131–73143, 2019, doi: 10.1109/ACCESS.2019.2920776.
- [35] D. Kurniadi, E. Fernando, S. Al Zayyan, and A. Mulyani, "Combined acoustic features with CNN–BiLSTM–Transformer for female emotion recognition," *Ingénierie des Systèmes d'Inf.*, vol. 30, no. 10, pp. 2727–2737, Oct. 2025, doi: 10.18280/isi.301018.
- [36] O. Alkadi, N. Moustafa, and B. Turnbull, "A collaborative intrusion detection system using deep blockchain framework for securing cloud networks," *Adv. Intell. Syst. Comput.*, vol. 1250, no. 1, pp. 1–15, 2021, doi: 10.1007/978-3-030-55180-3_41.
- [37] L. Yu, X. Yang, H. Wei, J. Liu, and B. Li, "Driver fatigue detection using PPG signal, facial features, head postures with an LSTM model," *Heliyon*, vol. 10, no. 21, pp. 1–10, Nov. 2024, doi: 10.1016/j.heliyon.2024.e39479.
- [38] D. Kurniadi, E. Fernando, A. Fauziyah, and A. Mulyani, "Improving low-light face recognition using DeepFace embedding and multi-layer perceptron," *J. RESTI (Rekayasa Sist. dan Teknol. Informatika)*, vol. 9, no. 5, pp. 1047–1055, Oct. 2025, doi: 10.29207/resti.v9i5.6797.

-
- [39] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," *Synth. Lect. Comput. Vis.*, vol. 8, no. 1, pp. 1–207, Feb. 2018, doi: 10.2200/S00822ED1V01Y201712COV015.
- [40] S. Swapna Rani, "Enhancing facial recognition accuracy in low-light conditions using convolutional neural networks," *J. Electr. Syst.*, vol. 20, no. 5s, pp. 2140–2148, 2024, doi: 10.52783/jes.2559.
- [41] M. R. Shafie, H. Khosravi, S. Farhadpour, S. Das, and I. Ahmed, "A cluster-based human resources analytics for predicting employee turnover using optimized artificial neural networks and data augmentation," *Decis. Anal. J.*, vol. 11, no. 1, pp. 1–12, 2024, doi: 10.1016/j.dajour.2024.100461.
- [42] V. Teodorescu and L. Obreja Braşoveanu, "Assessing the validity of k-fold cross-validation for model selection: Evidence from bankruptcy prediction using random forest and XGBoost," *Computation*, vol. 13, no. 5, pp. 1–12, May 2025, doi: 10.3390/computation13050127.
- [43] M. Lu, L. T. Tay, and J. Mohamad-Saleh, "Landslide susceptibility analysis using random forest model with SMOTE-ENN resampling algorithm," *Geomatics, Nat. Hazards Risk*, vol. 15, no. 1, pp. 1–15, 2024, doi: 10.1080/19475705.2024.2314565.
- [44] S. U. Natchiar and S. Baulkani, "Customer relationship management classification using data mining techniques," *Int. Conf. Sci. Eng. Manage. Res.*, vol. 2014, no. Nov., pp. 1–6, Nov. 2014, doi: 10.1109/ICSEMR.2014.7043662.
- [45] M. Salmi, D. Atif, D. Oliva, A. Abraham, and S. Ventura, "Handling imbalanced medical datasets: Review of a decade of research," *Artif. Intell. Rev.*, vol. 57, no. 10, pp. 1–40, Oct. 2024, doi: 10.1007/s10462-024-10884-2.
- [46] C. Llatas, B. Soust-Verdaguer, L. C. Torres, and D. Cagigas, "Application of knowledge discovery in databases (KDD) to environmental, economic, and social indicators used in BIM workflow to support sustainable design," *J. Build. Eng.*, vol. 91, no. 1, pp. 1–15, Aug. 2024, doi: 10.1016/j.jobbe.2024.109546.
- [47] U. Fayyad, "Knowledge discovery in databases: An overview," *Relational Data Mining*, vol. 2001, no. 1, pp. 28–47, 2001, doi: 10.1007/978-3-662-04599-2_2.
- [48] A. Bifet, J. Davis, T. Krilavičius, M. Kull, E. Ntoutsis, and I. Žliobaitė, Eds., "Machine learning and knowledge discovery in databases: Research track," *Lect. Notes Comput. Sci.*, vol. 14947, no. 1, pp. 1–15, 2024, doi: 10.1007/978-3-031-70368-3.
- [49] M. Rodríguez-Ibanez, A. Casáñez-Ventura, F. Castejón-Mateos, and P.-M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms," *Expert Syst. Appl.*, vol. 223, no. 1, pp. 1–15, Aug. 2023, doi: 10.1016/j.eswa.2023.119862.
- [50] F. Fitroh and F. Hudaya, "Systematic literature review: Analisis sentimen berbasis deep learning," *J. Nas. Teknol. dan Sist. Inf.*, vol. 9, no. 2, pp. 132–140, Aug. 2023, doi: 10.25077/TEKNOSI.v9i2.2023.132-140.