



Convolutional Neural Network Model for Multi-Class Sentiment Analysis on Twitter with Deep Learning Approach

Ade Putra Erwanda^{1*}, Al Khaidar², Nurdin³

^{1,2,3} Master of Information Technology Program, Malikussaleh University, Batam Street, Bukit Indah Campus, Lhokseumawe, Aceh.
adeputraerwanda.1998@gmail.com^{1*}, alkhaidarkuablang@gmail.com², nurdin@unimal.ac.id³

Abstract

Social media, especially Twitter, has become an important space for people to express opinions, emotions, and views on various issues. Sentiment analysis on Twitter text is a relevant approach to understanding people's emotional tendencies. However, most previous studies still focus on binary classification, making it less able to describe expressions. This study aims to implement a Convolutional Neural Network (CNN) model in multi-class sentiment analysis on Twitter data using a deep learning approach. The dataset used was obtained from Hugging Face with 11,324 entries in the form of text (tweets) without sentiment labels. The research stages include text preprocessing, application of lexicons for initial labeling, construction of a CNN architecture using embedding layers, convolutional layers, global max pooling, and dense layers with softmax activation. The model was compiled using categorical crossentropy as a loss function and the Adam optimizer, then drilled and tested using training and test data. The results show that the CNN is able to achieve a total accuracy of 97% with high precision, recall, and f1-score in almost all emotion classes. In terms of efficiency, the model requires only 37.96 seconds of execution time and uses 0.106 GB of memory, making it lightweight and efficient. Thus, CNN has proven effective in performing multi-class sentiment analysis on Twitter and has the potential to serve as a benchmark in the development of deep learning-based public opinion analysis systems.

Keywords: Convolutional Neural Network, Deep Learning, Sentiment Analysis, Twitter.

1. Introduction

The increasingly rapid development of information and communication technology has made social media an important part of modern society's life [1]-[2]. Twitter, as one of the popular social media platforms, is not only used to share information quickly, but also becomes a space for people to convey opinions, expressions, and sentiments on various issues [3]-[4]-[5]. This makes Twitter a rich source of data for analysis, especially in understanding the tendencies of public attitudes and emotions. Sentiment analysis is a widely used approach to extract meaning from text generated by social media users [6]-[7]. With sentiment analysis, text data can be categorized into specific classes, such as positive, negative, or neutral. However, as needs evolve, sentiment analysis is no longer sufficient with binary classification; instead, multi-class analysis is required to describe sentiment in more detail and complexity, resulting in more accurate and useful analysis results [8]. In the realm of artificial intelligence, machine learning methods have been widely applied in sentiment analysis [9]. However, conventional methods often face limitations in extracting features from text that are unstructured and diverse. Therefore, deep learning approaches have emerged that offer better capabilities in automatically learning text data representations without the need for manual feature extraction. One deep learning model that has proven effective in natural language processing is the Convolutional Neural Network (CNN), due to its ability to capture local patterns in text through a convolutional layer mechanism [10].

Although extensive research on sentiment analysis has been conducted, most still focuses on binary classification or traditional machine learning methods. Research examining multi-class sentiment analysis using CNNs, particularly on Twitter data, is still relatively limited. Applying CNNs in a multi-class context can provide a more in-depth view of the diversity of public opinion. Based on these issues, this study proposes a model (CNN) for multi-class sentiment analysis on Twitter using a deep learning approach. This research is expected to

contribute to the development of more accurate and adaptive sentiment analysis methods, as well as provide a broader understanding of communication and expression patterns on social media.

2. Literature Review

2.1. Deep Learning

Deep Learning is a branch of machine learning that focuses on the use of artificial neural networks with many layers (deep neural networks) to learn complex data representations [11]. With its ability to automatically extract features, deep learning is widely used in the fields of image processing, speech recognition, and natural language processing. Deep learning models such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) have been proven to be able to overcome the limitations of conventional methods in handling large and unstructured data [12].

2.2. Sentiment Analysis

Sentiment analysis is a branch of text mining and natural language processing that aims to identify, extract, and classify opinions or emotions contained in text [13]. In general, sentiment analysis can be divided into three categories, namely binary analysis (positive/negative), multi-class analysis (e.g., very positive, positive, neutral, negative, very negative), and aspect-based sentiment analysis. In the context of this research, the focus is directed at multi-class sentiment analysis to gain a deeper understanding of variations in public opinion.

2.3. Twitter

Twitter is one of the most widely used social media platforms for expressing opinions, sharing information, and interacting with other users in real-time. Twitter's main characteristic is the character limit per tweet, which encourages users to convey short but concise messages [14]. In addition, the use of hashtags (#), mentions (@), and retweets makes Twitter a dynamic, meaningful, and relevant source of text data for analysis. Due to its open nature, Twitter data is often used in sentiment analysis research, public opinion mapping, and detecting social issue trends.

2.4. Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning architecture that was initially widely used in image processing, but later developed and applied to natural language processing. CNN works by applying convolution operations to extract local features from data, as well as utilizing pooling layers to reduce data dimensions without losing important information. In the context of text analysis, CNN is able to detect significant word and phrase patterns in determining sentiment. The advantage of CNN is its ability to capture spatial and contextual representations of text data, making it effective for sentiment analysis, including in cases of multi-class classification. The steps for solving Convolutional Neural Network are [15].

1. Input Data: Twitter text is processed and converted into a numerical representation (embedding).
2. Convolution Layer: A moving filter extracts important patterns from the text (e.g., word combinations).
3. Activation Function (ReLU): Adds non-linearity so the model can learn more complex patterns.
4. Pooling Layer: Reduces data size while retaining the most important features.
5. Flattening: The pooling results are converted into a one-dimensional vector.
6. Fully Connected Layer: Combines the extracted features for the classification process.
7. Output Layer (Softmax): Generates probabilities for each sentiment class (multi-class).

3. Research Methodology

3.1 Data Collection

Data collection is a crucial initial step in this research, as data quality directly impacts the performance of the model being built. The data used in this study was obtained from Hugging Face, a platform widely used by machine learning researchers and practitioners worldwide for sharing models and datasets. Hugging Face was chosen as the data source for several reasons.

1. Availability of a standardized dataset, making it easier to use in academic research and practical implementation.
2. Good data quality, as the dataset has been curated by the community and competent researchers.
3. High accessibility, as the dataset can be downloaded and used for free and supports various processing formats.
4. Relevance to current research, as Hugging Face is a leading reference in the field of Natural Language Processing (NLP).

A total of 11,324 data entries were collected, which were then used to train and test a Convolutional Neural Network (CNN) model for multi-class sentiment analysis.

3.2. Research Process Flow

The research process began with the collection of 11,324 entries of data from Hugging Face. The data then underwent preprocessing for text cleaning and normalization, followed by lexicon labeling to ensure each text had a corresponding emotion class. Data was then split into training and test data. In the model training stage, the training data was used to train a CNN model through several layers until it was able to extract important features. Model testing and classification were then performed with the test data to measure model performance.

Test results were evaluated using accuracy, precision, recall, and f1-score metrics during the evaluation stage, and visualization was performed in the final stage to display the classification results in graphs and performance tables. The flow can be seen in Figure 1.

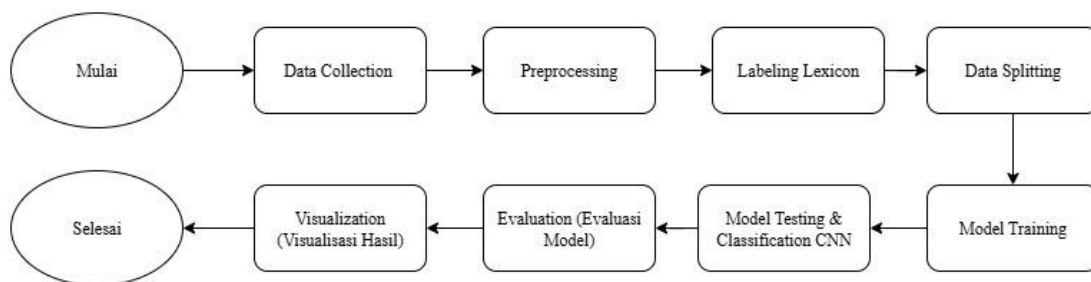


Fig. 1: Research Process Flow

4. Result and Discussion

This chapter discusses the research findings obtained through the application of a Convolutional Neural Network (CNN) model to multi-class sentiment analysis on Twitter data. The discussion focuses on interpreting the data processing results, evaluating model performance, and comparing them with previous research. Furthermore, this section explains the implications of the research findings for understanding public opinion on social media and their contribution to the development of deep learning-based sentiment analysis methods.

4.1. Data Collection

The data used in this study was collected from Hugging Face as the dataset source. The data obtained amounted to 11,324 entries in the form of raw text (tweets) without sentiment labels. Because the data is still unclassified, a data labeling process is required in the next stage so that each text has an appropriate sentiment category. The Hugging Face dataset was selected based on its open availability, structured, and frequent use in research related to text analysis, thus supporting the validity and enabling this study to be replicated. The data collection data can be seen in Figure 2.

```

    Jumlah data: 11324
                                     text
0  bubuk ayam yang lumayan rekomendasi di sekitar...
1  menu bebek relatif jarang di bandung dan bebek...
2  hampir lebih 5 kali saya ke sini . dim sum nya...
3  tempat nya dekat dengan factory outlet jadi ha...
4  saya tidak sengaja menemukan warung bakso dan b...
..  ...
95 sejak pindah nya fungiyaki ramen dan sushi bar...
96                                     tidak pantas dipercaya
97 yang menjadi ciri khas di easton sebenarnya ku...
98 saya datang ke sini 2 kali bersama teman-teman...
99 ganjal perut saat ingin makan cepat . lokasi m...
  
```

Fig. 2: Data Collection

Figure 2 shows a collection of raw data taken from Hugging Face with a total of 11,324 data. The data is still in the form of text (tweets) without sentiment labels, so it requires a preprocessing and labeling stage before it can be used in the sentiment analysis process using the Convolutional Neural Network (CNN) model.

4.2 Text Preprocessing

Before data is used in model training, a text preprocessing stage is required to ensure the quality of the data to be analyzed. This stage aims to clean the text of irrelevant elements, standardize the format, and transform the raw data into a form suitable for processing by a Convolutional Neural Network (CNN) model. Preprocessing makes the data more structured and representative, thus improving the model's performance in multi-class sentiment analysis. The results of the text preprocessing can be seen in Figure 3.

```

                                     clean_text
0  bubuk ayam yang lumayan rekomendasi di sekitar...
1  menu bebek relatif jarang di bandung dan bebek...
2  hampir lebih 5 kali saya ke sini dim sum nya yah...
3  tempat nya dekat dengan factory outlet jadi ha...
4  saya tidak sengaja menemukan warung bakso dan b...
  
```

Fig. 3: Text preprocessing results

4.3 Application of Lexicon

At this stage, a lexicon is applied as an initial reference in the sentiment analysis process. The lexicon is used to provide information about the polarity of words, whether they are positive, negative, or neutral. By applying the lexicon to text data, each word or phrase in a tweet can be mapped to a specific sentiment value. This process helps enrich the data representation before entering the Convolutional Neural Network (CNN) model training stage and serves as the basis for determining the direction of multi-class sentiment classification. The results of applying the lexicon can be seen in Figure 4.

	clean_text	label
0	bubur ayam yang lumayan rekomendasi di sekitar...	Takut
1	menu bebek relatif jarang di bandung dan bebek...	Sedih
2	hampir lebih kali saya ke sini dim sum nya yah...	Gembira
3	tempat nya dekat dengan factory outlet jadi ha...	Sedih
4	saya tidak sengaja menemuka warung bakso dan b...	Takut
5	makanan di rs selalu tidak enak	Gembira
6	walau lokasi cukup sulit untuk dijangkau dusun...	Gembira
7	tidak iri	Takut
8	kasihan personal ungu sekarang sepi job gara g...	Gembira
9	ulasan singkat tipe mal makan malam buffett lo...	Takut
10	suasana nya asik romantis dan eksklusif pemanda...	Marah
11	kafe dan restorant the jam sibuk lokasi nya di...	Sedih
12	bangga deh punya presiden kayak pak jokowi yan...	Gembira
13	ri ku molo adalah orang pdip yang di partai ny...	Marah
14	minggu ini internet indihome saya terganggu da...	Takut
15	kesal saya melihat manusia satu ini	Marah
16	heran ya rata rata pendukung ahok ini seperti ...	Gembira
17	pakai mobil ford yang kata nya irit nyata nya ...	Marah
18	terima kasih id aku dapat voucher tidak menyese...	Gembira
19	lokasi kafe ini di bandung atas dago dengan lo...	Marah

Fig. 4: Lexicon

Figure 4 shows the process of applying a lexicon to text data, where each word in a tweet is mapped to a sentiment category based on a predetermined polarity. This stage serves to provide an initial overview of the sentiment distribution before the data is further processed using a Convolutional Neural Network (CNN) model.

4.4. Implementation of the Convolutional Neural Network Model

In this stage, a Convolutional Neural Network (CNN) model was implemented for multi-class sentiment analysis on Twitter data. The model was built using a sequential approach with several main layers: an Embedding layer to convert text into vector representations, a Convolutional layer (Conv1D) to extract local features from the text data, GlobalMaxPooling1D to reduce feature dimensionality, and a Dense layer with softmax activation in the output layer to generate sentiment predictions based on the number of classes.

The model implementation process included building the CNN architecture, compiling the model with the categorical crossentropy loss function and the Adam optimizer, training the model using training data, and evaluating model performance using test data. The results of this stage serve as the basis for measuring the accuracy of the CNN in multi-class sentiment classification. The results of the model implementation can be seen in Figure 5.

```
# 8. Bangun model CNN
model = Sequential()
model.add(Embedding(input_dim=5000, output_dim=128, input_length=100))
model.add(Conv1D(filters=128, kernel_size=5, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(y.shape[1], activation='softmax')) # jumlah kelas sesuai Label

# 9. Compile
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# 10. Train
history = model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))

# 11. Evaluasi
loss, acc = model.evaluate(X_test, y_test)
print(f"Akurasi: {acc*100:.2f}%")
```

Fig. 5: Implementation of Convolutional Neural Network

4.5 Sentiment Prediction

This stage is the sentiment prediction process using a pre-trained Convolutional Neural Network (CNN) model. Preprocessed test data is fed into the model to generate probabilities for each sentiment class. These probabilities are then determined by the dominant class using the softmax function, allowing each tweet to be classified into the appropriate sentiment category. Thus, this prediction stage provides an overview of the model's ability to recognize text patterns and accurately classify multi-class sentiment. The sentiment prediction results can be seen in Figure 6.

```
Teks: saya di marahin sama ade, padahal saya tidak salah => Prediksi Emosi: Marah
Teks: aku gelap dan sendirian => Prediksi Emosi: Netral
Teks: saya gembira hari ini karena lulus => Prediksi Emosi: Gembira
Teks: aku sedih ditinggalkan sahabatku => Prediksi Emosi: Sedih
```

Fig. 6: Emotional Sentiment Prediction

4.6 Visualization of Model Performance in Training and Validation

To better understand model performance, we visualized model performance during the training and validation processes. This visualization includes a comparison of accuracy and loss values on the training data and validation data at each epoch. The visualization allows us to observe model learning patterns, indications of overfitting or underfitting, and the stability of CNN performance in multi-class sentiment classification. The model performance results can be seen in Figure 7.

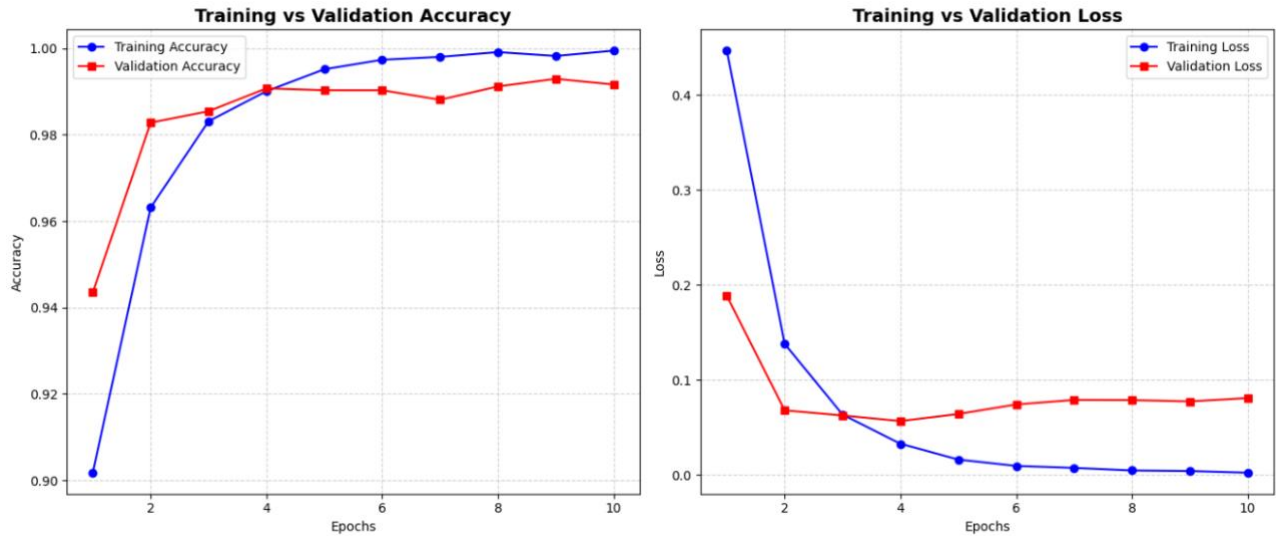


Fig. 7: Visualization of Model Performance in Training and Validation

Figure 7 shows a visualization of the performance of the Convolutional Neural Network (CNN) model during the training and validation processes. The graph shows that the final accuracy value on the training data reached 99.94%, while the final accuracy on the validation data reached 99.16%. This indicates that the model has a very high level of accuracy in recognizing patterns in both the training and test data. Meanwhile, in terms of error (loss), the final loss obtained on the training data was 0.0026 and the final loss on the validation data was 0.0811. The relatively small difference in loss values between training and validation indicates that the model did not experience significant overfitting and was able to generalize well on previously unseen data.

4.7 Visualization of Convolutional Neural Network Performance Per Class

This section displays the visualization of the Convolutional Neural Network (CNN) model performance per class. This visualization aims to determine the model's ability to accurately classify each sentiment category. This visualization provides a detailed overview of the prediction distribution, accuracy level, and errors within each class, thus helping to evaluate model performance more thoroughly. The results of the Convolutional Neural Network Performance Visualization per Class can be seen in Figure 8.

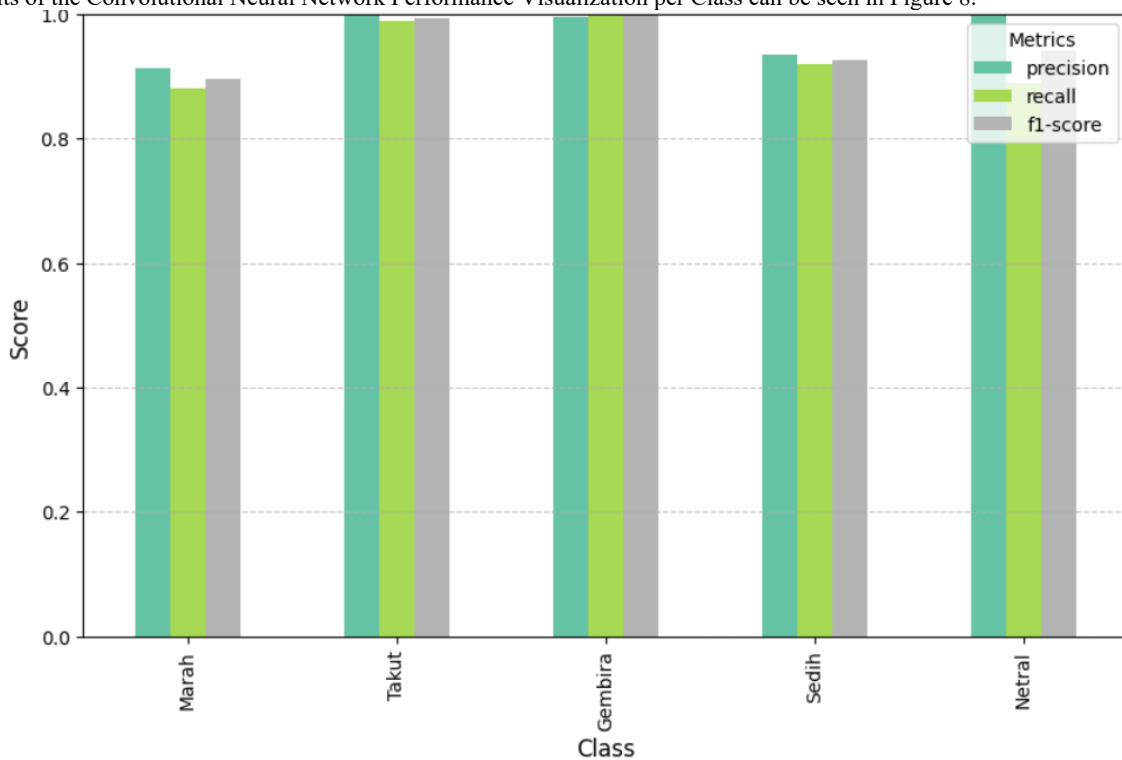


Fig. 8: Visualization of Convolutional Neural Network Performance Per Class

Figure 8 shows the visualization of the CNN model's performance in classifying sentiment based on each emotion class. The table shows that the Angry class achieved a precision of 0.91, a recall of 0.88, and an f1-score of 0.90, indicating a fairly good model performance, although some prediction errors were present. For the Fear class, the model achieved very high performance with a precision of 1.00, a recall of 0.99, and an f1-score of 0.99, indicating almost no classification errors. The Happy class even achieved perfect results with a precision, recall, and f1-score of 1.00 each, indicating the model was able to recognize this class without error. For the Sad class, a precision of 0.93, a recall of 0.92, and an f1-score of 0.93 also demonstrated good performance. Meanwhile, for the Neutral class, the precision

reached 1.00, but the recall was only 0.89, resulting in an f1-score of 0.94. Overall, the CNN model achieved a total accuracy rate of 97%, which proves that this method is capable of performing multi-class emotion classification with a very high level of reliability.

4.8 Computational Efficiency and Feasibility

The computational efficiency and feasibility of the Convolutional Neural Network (CNN) model used in the study were evaluated. The analysis was conducted by considering training time, computational resource usage, and model complexity relative to dataset size. This evaluation is crucial to assess the model's ability to not only achieve high accuracy but also to run effectively and efficiently in the available computing environment.

Table 1: Computational Cost

Computational Cost (CNN)	
Execution Time	37.96 seconds
Memory Usage	0.106 GB

Table 1 shows the test results, indicating that the CNN model had an execution time of 37.96 seconds during the training and evaluation process. This indicates that the model can be run relatively quickly on a dataset of 11,324 entries, thus supporting computational efficiency. Furthermore, the memory usage of only 0.106 GB indicates that this model is relatively lightweight and does not require large hardware resources. Thus, CNN is proven not only to be accurate in performing multi-class classification, but also efficient and feasible for implementation in systems with limited computing resources.

5. Conclusion

This study successfully implemented a Convolutional Neural Network (CNN) model for multi-class sentiment analysis on 11,324 Twitter entries obtained from Hugging Face. Through text preprocessing, lexicon application, and CNN model training, the study demonstrated that the CNN method is capable of extracting important features from text data and producing excellent classification performance. Test results demonstrated a total accuracy rate of 97%, with high precision, recall, and f1-score values across nearly all emotion classes. This demonstrates CNN's robust ability to recognize text patterns and perform multi-class sentiment classification with high accuracy.

Furthermore, in terms of computational efficiency and feasibility, the developed CNN model demonstrated optimal performance. With an execution time of only 37.96 seconds and memory usage of 0.106 GB, this model is lightweight and practical for implementation, even on systems with limited computing resources. Overall, this research not only contributes to the application of CNN for multi-class sentiment analysis, but also proves that this method is effective, efficient, and worthy of being used as a reference in the development of a public opinion analysis system in social media based on deep learning.

References

- [1] A. Khaidar, M. Arhami, and M. Abdi, "Application of the Random Forest Method for UKT Classification at Politeknik Negeri Lhokseumawe," *Journal of Artificial Intelligence and Software Engineering*, vol. 4, no. 2, pp. 94–103, 2024.
- [2] G. A. de Oliveira Júnior, R. de Oliveira Albuquerque, C. A. Borges de Andrade, R. T. de Sousa Jr, A. L. Sandoval Orozco, and L. J. García Villalba, "Anonymous Real-Time Analytics Monitoring Solution for Decision Making Supported by Sentiment Analysis," *Sensors*, vol. 20, no. 16, p. 4557, Aug. 2020, doi: 10.3390/s20164557.
- [3] A. Vishwakarma and M. Chugh, "COVID-19 vaccination perception and outcome: society sentiment analysis on twitter data in India," *Social Network Analysis and Mining*, vol. 13, no. 1, p. 84, May 2023, doi: 10.1007/s13278-023-01088-7.
- [4] S. Alghamdi and Y. Alhasawi, "Aspect-based sentiment analysis in smart devices: A comprehensive and specialized dataset," *Data in Brief*, vol. 55, p. 110642, Jun. 2024, doi: 10.1016/j.dib.2024.110642.
- [5] Z. T. Osakwe, I. Ikhapoh, B. K. Arora, and O. M. Bubu, "Identifying public concerns and reactions during the COVID-19 pandemic on Twitter: A text-mining analysis," *Public Health Nursing*, vol. 38, no. 2, pp. 145–151, Mar. 2021, doi: 10.1111/phn.12843.
- [6] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2021.
- [7] A. TiaraSari and E. Haryatmi, "Penerapan Convolutional Neural Network Deep Learning dalam Pendeteksian Citra Biji Jagung Kering," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 265–271, 2021.
- [8] B. Siswoyo and N. A. P. Utomo, "Pemanfaatan Machine Learning untuk Klasifikasi Sentimen Pelanggan pada Media Sosial," *Jurnal Teknologi Informasi Digital*, vol. 1, no. 1, pp. 29–34, 2025.
- [9] I. W. Suasnawa and I. G. N. B. Caturbawa, "Penerapan Machine Learning untuk Analisis Sentimen di Media Sosial: Sebuah Tinjauan Literatur Sistematis," *Ganapati - Jurnal Rekayasa Teknik*, vol. 3, no. 1, pp. 36–40, 2025.
- [10] S. D. W. Pratama, F. Panjaitan, M. I. Herdiansyah, S. D. Purnamasari, and N. A. O. Saputri, "Penerapan Metode Machine Learning Naive Bayes Pada Twitter Terhadap Sentimen Pada Pembelajaran Daring Di Indonesia," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 6, pp. 11684–11688, 2024.
- [11] Z. Xi, F. Zhuang, G. Zhou, X. Cheng, F. Lin, and Q. He, "Domain Adaptation with Category Attention Network for Deep Sentiment Analysis," *arXiv preprint arXiv:2112.15290*, Dec. 2021.
- [12] C. N. Dang, "Hybrid Deep Learning Models for Sentiment Analysis," *Complex & Intelligent Systems*, vol. 7, pp. 1713–1723, 2021, doi: 10.1155/2021/9986920.
- [13] I. Priyadarshini, P. K. Sahu, R. Mallick, A. K. Sangaiah, and M. S. Hossain, "A novel LSTM–CNN–grid search-based deep neural network model for sentiment analysis," *IEEE Access*, vol. 9, pp. 106907–106917, 2021, doi: 10.1109/ACCESS.2021.3099012.
- [14] A. Alhadlaq and A. Altheneyan, "A new hybrid deep learning approach for aspect-based sentiment analysis," *PeerJ Computer Science*, vol. 10, p. e2267, 2024, doi: 10.7717/peerj-cs.2267.
- [15] Y. Chen, J. Fan, and D. Wang, "Local interpretation of deep learning models for Aspect-Based Sentiment Analysis," *Engineering Applications of Artificial Intelligence*, vol. 132, p. 107612, 2024, doi: 10.1016/j.engappai.2024.107612.