INTENSIF, Vol.8 No.2 August 2024
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

Sentiment Analysis of Sirekap Tweets Using CNN Algorithm

Received:
8 July 2024
Accepted:
20 August 2024
Published:
31 August 2024

[1*]Handoko, [2]Ahmad Asrofiq, [3]Junadhi, [4]Ari Sukma Negara
[1-3]Teknik Informatika, Universitas Sains & Teknologi Indonesia
[4]Master of Data Science, University Texas at Austin
E-mail: [1]handoko0193@gmail.com [2]asrofiqtr12@gmail.com, [3]junadhi@sar.ac.id, [4]ari.sukmanegara@gmail.com
*Corresponding Author

Abstract— Background: The research investigates the application of deep learning models for sentiment analysis on Twitter data related to Indonesia's Sirekap system. Sentiment analysis is crucial for understanding public opinion and enhancing the transparency and reliability of election result recapitulation processes. **Objective**: The objective of this study is to compare the performance of Convolutional Neural Networks (CNN) and CNN-LSTM models in analyzing sentiments from tweets about the Sirekap system. The study aims to identify the most effective model and preprocessing techniques to improve sentiment classification accuracy. **Methods**: A comprehensive data preprocessing pipeline was implemented, including cleansing, case folding, tokenizing, normalization, stopword removal, and stemming. To address class imbalance, the SMOTE technique was applied. The models were trained and evaluated using accuracy, precision, recall, and F1-score metrics. Pre-trained word embeddings were used to enhance model performance. **Results:** The CNN model achieved an accuracy of 85.90%, outperforming the CNN-LSTM model, which achieved 79.91% accuracy. Additionally, the CNN model demonstrated superior precision, recall, and F1-score metrics compared to the CNN-LSTM model. The thorough preprocessing and handling of class imbalance significantly contributed to the enhanced performance of the CNN model. **Conclusion**: The research emphasizes the effectiveness of deep learning approaches, particularly CNNs, in sentiment analysis tasks. The findings highlight the importance of comprehensive preprocessing and class imbalance handling. The use of pre-trained word embeddings and various evaluation metrics ensures robust model performance. These insights contribute to improving the accuracy and efficiency of sentiment classification, thereby enhancing the reliability and transparency of election result recapitulation processes.

Keywords— Sentiment Analysis; Deep Learning; Convolutional Neural Networks (CNN); CNN-LSTM; SMOTE; Sirekap System

Corresponding Author:
Handoko,
Teknik Informatika,
Universitas Sains & Teknologi Indonesia,
handoko0193@gmail.com
Orchid ID: https://orcid.org/0009-0002-6791-7756

INTENSIF, Vol.8 No.2 August 2024
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

I. INTRODUCTION

Indonesia has conducted many elections since its independence in 1945 [1]. The General Election Commission (KPU) is an institution tasked with organizing general elections in Indonesia. Based on Law Number 7 Year 2017, the KPU has the duty and authority to organize general elections to elect the president and vice president, members of the DPR, members of the DPD, and members of the DPRD. KPU determines the permanent voter list, the number of seats for political parties participating in the election, campaign time limits, election regulations, vote counting procedures, and polling station locations. KPU compiles and determines the recapitulation of general election results, and carries out other duties and authorities granted by law. KPU is responsible for carrying out its duties and authorities in a professional, transparent and accountable manner in order to create clean and honest elections [2].

The Recapitulation Information System is called sirekap. Indonesia's General Election Commission (KPU) uses this information system to assist in the counting and recapitulation of general and regional election results. The system enables the collection of vote data from the lowest level to the national level, allowing for a faster, more effective and transparent vote counting process. Sirekap helps KPU conduct recapitulation electronically, releasing them from reliance on manual processes that can be faulty and manipulated. As a result, Sirekap is expected to improve the reliability and accuracy of election results and increase public confidence in the integrity of the electoral process [3].

Sentiment analysis, also known as opinion mining, is a procedure that aims to identify and categorize the different types of emotions or opinions that users may have in relation to a particular service, product, event, or feature. The main goal is to find out whether the opinions are positive, negative, or neutral. In this case, the emotions or opinions can cover various aspects, such as user satisfaction, product quality, policies, and so on [4], [5], [6].

Deep Learning can be used for sentiment analysis. One commonly used approach is to use neural network architectures such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), especially Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU), to analyze text and extract relevant features to predict sentiment [7]. Convolutional neural networks (CNNs), which are extensively employed in natural language processing domains including text classification and sentiment classification, are utilized in this study due to their potent feature learning capabilities and feature representation [8].

This research aims to understand the application of sentiment analysis regarding public perceptions related to elections in Indonesia, specifically concerning the use of Sirekap by the General Election Commission (KPU). The focus of this study is to analyze public sentiment [9],

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

compare CNN and LSTM models with traditional methods, and provide recommendations to enhance public trust in the electoral process. The methodology employed includes social media data analysis as well as the implementation and evaluation of CNN and LSTM models.

The significance of this research lies in its ability to provide insights into the use of deep learning in political sentiment analysis. It is expected that the findings will help KPU understand public perceptions of technologies like Sirekap, enabling more informed decision-making to improve transparency and trust in the electoral process. Additionally, the analysis results can provide a foundation for developing policies that are more responsive to public opinion, which in turn can strengthen the legitimacy of the electoral process in Indonesia. By offering a better understanding of public sentiment, this research has the potential to enhance public trust in both technology and the electoral process, which is essential for the stability of democracy.

A related study by P.O. Abas Sunarya compared the accuracy of Convolutional Neural Networks (CNN) and Naïve Bayes classifier in sentiment analysis of Twitter data. The CNN classification model achieved 89% accuracy in the training and evaluation process, and 88% in the testing process. However, the Naïve Bayes classification model had an accuracy of 78%. These results show that the CNN classification model has better accuracy than the Naïve Bayes classification model [10].

An investigation carried out by Gusti Agung Mayun Kukuh Jaluwana, In this work, pre-trained word embeddings from fastText are used to build Deep Learning models using 1,875 training data sets. The Deep Learning approach differs in its ability to forecast public opinion regarding the government's initiatives to break the COVID-19 transmission chain. 94.40% accuracy, 94.39% precision, 94.40% recall, and 94.39% F1-score were attained via CNN architecture. In the meantime, 97.34% accuracy, 97.31% precision, 97.33% recall, and 97.31% F1-score were attained by the Bidirectional LSTM Architecture. Because it can handle both short- and long-term information bidirectionally and has a forgetting door to identify pertinent information, bidirectional LSTM is superior [11].

According to research done by Muhammad Dehghani, This work uses deep learning (CNN-LSTM) and a number of machine learning techniques (Gaussian Naive Bayes, Gradient Boosting, Logistic Regression, Decision Trees, and Random Forests) to evaluate sentiment analysis on political tweets written in Persian. ParsBERT was used for deep learning and Bag of Words (BOW) for machine learning in the classification of the tweets. The evaluation was conducted using two datasets from Twitter: one with three classes and the other with seven classes. The study's objectives of shortening training times and accelerating model speed were accomplished. CNN-LSTM produced the best accuracy, recall, and F1 scores (89%, 89%, and 88%),

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

respectively, on the first dataset. Furthermore, CNN-LSTM obtained the highest ratings for all evaluation criteria on the second dataset, with 71% scores for accuracy, precision, and recall [12].

Furthermore, research conducted by Sio Journalist Pipin, this research explores public sentiment about ChatGPT in Indonesia, especially on Twitter. Using deep learning-based sentiment analysis, this research maps diverse opinions about ChatGPT, ranging from praise for technological advances to criticism of its potential adverse effects. Data analysis showed significant differences between CNN and Fast R-CNN in classifying sentiments from Indonesian tweets. The dataset consists of 7,604 tweets with "Positive", "Negative", and "Neutral" sentiments. The Fast R-CNN model showed an accuracy rate of 94.5%, higher than the CNN model which reached 86%. These findings demonstrate the potential of Fast R-CNN in sentiment analysis to unearth deeper nuances from Twitter data in Indonesia, providing a richer understanding of people's perceptions of ChatGPT [13].

II. RESEARCH METHOD

Figure 1 illustrates the study methodology employed at multiple stages. The process begins with a literature review, followed by data collection from social media platforms such as Twitter. Next, the gathered text is preprocessed, and training datasets are collected and manually labeled. Afterward, text vectorization is performed, and a model architecture is built and trained. The best model is then evaluated using a confusion matrix. The process continues with data prediction using all collected datasets, including the training data, and concludes with the visualization of results.

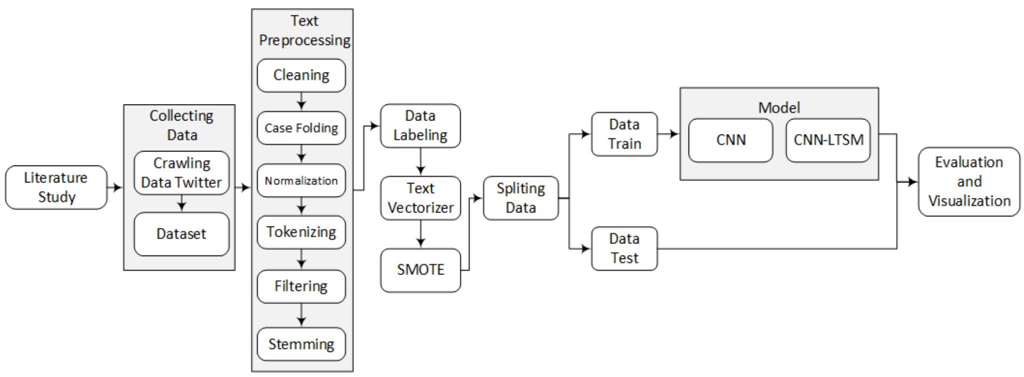

Fig 1. Reseacrh Method

INTENSIF, Vol.8 No.2 August 2024
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

A. Collecting Data

The data used in this research consists of tweets from the social media platform Twitter, where many Indonesians express their opinions about Sirekap. The process employed for data collection is data crawling [14]. The keyword used for this research is "Sirekap," and the data was collected from December 2023 to January 2024. As shown in Figure 2, the crawling process yielded a total of 2,518 tweets, with the dataset containing only the date and the tweet content.

	created_at	full_text
0	Wed Jan 31 23:46:02 +0000 2024	Edisi pagi-pagi dah sambat soal sirekap wkkwkw...
1	Wed Jan 31 23:42:47 +0000 2024	@annonsx sirekap kan ga support ios wkwkwk
2	Wed Jan 31 23:41:12 +0000 2024	Siapa si sirekap sirekap itu?? Kok bikin aku ...
3	Wed Jan 31 23:32:55 +0000 2024	Udah dibilang hp gue gak support si rekap, mas...
4	Wed Jan 31 23:32:32 +0000 2024	@kryuk28 haloo. buat yg belom ada notif dari k...
...	...	...
2513	Thu Dec 14 08:20:37 +0000 2023	#TemanPemilih, Divisi Sosialisasi, Pendidikan ...
2514	Thu Dec 14 05:50:23 +0000 2023	Kalawat - Menyiapkan hari pemungutan suara KPU...
2515	Thu Dec 14 03:24:24 +0000 2023	KPU Kab. Gowa menggelar Rapat Koordinasi Optim...
2516	Thu Dec 14 13:15:41 +0000 2023	#TemanPemilih KPU Kabupaten Pasuruan gelar Bim...
2517	Wed Dec 13 17:05:24 +0000 2023	Selain itu, M. Agus Muslim juga memberikan per...

2518 rows × 2 columns

Fig 2. Result of Crawling

B. Text Preprocessing

Text preprocessing in sentiment analysis involves a series of steps designed to clean and prepare text data for proper processing by sentiment analysis algorithms [15], [16]. The preprocessing stage includes several processes: cleansing, case folding, tokenization, normalization, stopword removal, and stemming [17].

Firstly, the text is cleansed to eliminate undesired characters, such as symbols, italics, punctuation marks, URLs, and other unnecessary special characters [18], [19]. The goal of data cleansing is to enhance data quality and facilitate accurate sentiment analysis. Figure 3 below illustrates the results of the cleansing process.

Udah dibilang hp gue gak support si rekap, masih aja diminta jadi si rekap terus suruh tukeran hp sama yg lain. Lah? Aneehhh wkwk	→	Udah dibilang hp gue gak support si rekap masih aja diminta jadi si rekap terus suruh tukeran hp sama yg lain Lah Aneehhh

Fig 3. Result of Cleansing

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

Case folding is the second step in the preprocessing process. This stage is almost always included when preparing text, as inconsistent formatting can pose significant challenges for data management and analysis. To address this issue, capital letters are converted to lowercase through case folding [20]. Figure 3 below illustrates the results of the case folding process.

Udah dibilang hp gue gak support si rekap masih aja diminta jadi si rekap terus suruh tukeran hp sama yg lain Lah Aneehhh	→	udah dibilang hp gue gak support si rekap masih aja diminta jadi si rekap terus suruh tukeran hp sama yg lain lah aneehhh

Fig 4. Result of Case Folding

The Colloquial Indonesian Lexicon dictionary and the INANLP dictionary, accessed through GitHub, are critical components of the normalization stage in this research. During this process, both dictionaries are incorporated into Python code as data dictionary types. The underlying principle of this normalization process is that if each token in the tweet data matches a word in the dictionary, it will be converted to its base form [21], [22]. Figure 5 illustrates the results of the normalization process.

udah dibilang hp gue gak support si rekap masih aja diminta jadi si rekap terus suruh tukeran hp sama yg lain lah aneehhh	→	sudah dibilang hp saya tidak support si rekap masih saja diminta jadi si rekap terus suruh bertukar hp sama yang lain lah aneehhh

Fig 5. Result of Normalization

The next step is tokenization, which involves dividing the text into individual words or tokens. This process breaks down sentences into smaller components, making text processing easier by allowing for the identification of important units (tokens) [23], [24]. Figure 6 illustrates the results of the tokenization process.

sudah dibilang hp saya tidak support si rekap masih saja diminta jadi si rekap terus suruh bertukar hp sama yang lain lah	→	sudah, dibilang, hp, saya, tidak, support, si, rekap, masih, saja, diminta, jadi, si, rekap, terus, suruh, bertukar, hp, sama, yang, lain, lah, aneehhh

Fig 6. Result of Tokenization

INTENSIF, Vol.8 No.2 August 2024
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

After tokenization, the next step is filtering, which involves removing stop words and other irrelevant words. This strategy focuses on eliminating common words that do not contribute to sentiment analysis, thereby emphasizing more significant words. By doing so, we aim to achieve more accurate sentiment results [25]. Figure 7 illustrates the results of the filtering process.

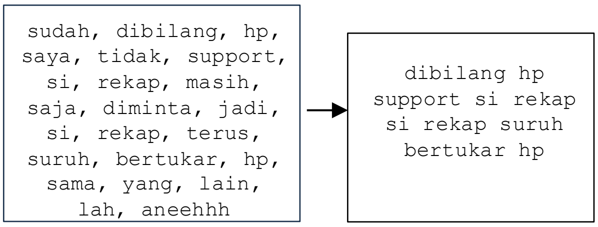

Fig 7. Result of Filtering

The final step is stemming, a process that converts words in a text into their base form, or root words [26], [27]. The goal of stemming is to remove various word endings and reduce words to their basic form, known as the "stem" or "root." Stemming is commonly used in natural language processing to reduce text complexity and facilitate text analysis, such as data retrieval and document classification.

While Porter's algorithm is widely used for English text, the Sastrawi algorithms are the most commonly used stemming algorithms for Indonesian [28]. In this research, we will utilize the Sastrawi algorithm. Figure 8 illustrates the results of the stemming process.

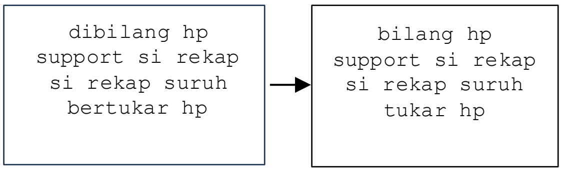

Fig 8. Result of Stemming

C. Labelling Data

The lexicon-based labeling method is a technique for sentiment analysis that utilizes a list of words (lexicon) with predefined sentiment values to identify sentiment within the analyzed text [29]. In this context, the labeling assigns positive and negative values to the text. Afterward, the data is filtered to meet specific criteria, eliminating content that lacks diverse viewpoints. As a result of this filtering process, 2,077 tweets were obtained that reflect the opinions expressed, with a total of 907 positive values and 1,170 negative values. Table 1 below presents the results of the manual labeling.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

Table 1. Result of Labeling Data

Tweet	Label
hai warga tedunan pps desa tedunan hadir simulasi mungut hitung suara simulasi guna sirekap kpu demak	positive
mending lembur kantor main hp bimtek sirekap pusing	Negative

D. Text Vectorization

In text processing, text vectorization is the process of converting unprocessed text into a numerical format that deep learning models can utilize [30]. This process transforms raw text into a representation suitable for deep learning algorithms by using numerical values, which is the essence of text vectorization.

E. SMOTE

A technique known as SMOTE (Synthetic Minority Over-sampling Technique) is employed to address the issue of class imbalance within datasets. A class is considered imbalanced when it has significantly fewer samples than other classes, which can lead to machine learning models that tend to overlook minority classes [31], [32]. By using SMOTE in deep learning, we can enhance the model's performance on unbalanced datasets and ensure that it makes more equitable predictions for minority classes.

F. Splitting Data

Dividing a dataset into different subsets for training and testing machine learning or deep learning models is known as data splitting. This is a crucial step in the machine learning workflow, as it ensures that the models built can be accurately evaluated and possess strong generalization abilities on unseen data.

In the scikit-learn library, the **train_test_split** function is commonly used to randomly split the data into training and testing sets. This technique allows you to specify parameters such as **train_size** and **test_size** to determine the size of these subsets. For instance, if **test_size** = 0.25, 25% of the data will be allocated to the testing set, while 75% will go to the training set. Additionally, the **random_state** parameter can be used to ensure the reproducibility of the split, and the stratify parameter helps maintain the same class distribution in both the training and testing sets [33].

G. Model

In this research, both CNN and CNN LSTM models will be utilized to determine which model achieves the highest accuracy. Convolutional Neural Networks (CNNs) are a type of artificial neural network architecture that is particularly effective for processing data with a grid-like structure, such as images. CNNs consist of multiple layers that function to extract features from

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

input data through a convolutional process [34], [35]. Figure 9 below illustrates the architecture of the CNN model.

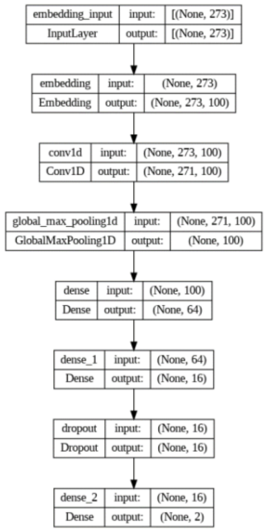

Fig 9. Architecture of CNN

CNN-LSTM is a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM) designed to handle data with both temporal and spatial properties. This hybrid model is crucial in various applications, such as text analysis, weather prediction, and video analysis, where understanding spatial patterns (through CNN) and temporal or sequential patterns (through LSTM) is essential [36], [37]. Figure 10 below illustrates the architecture of the CNN+LSTM model.

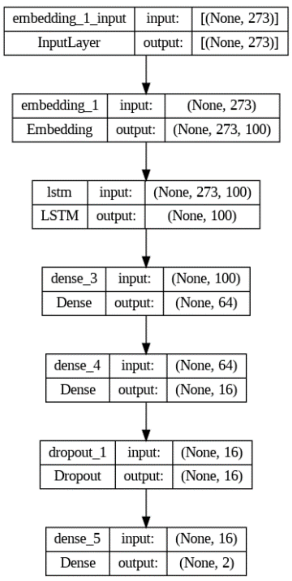

Fig 10. Architecture of CNN+LSTM

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

H. Evaluation and Visualization

Evaluation and visualization are crucial steps in the development of machine learning models. The bar graph illustrating the before-and-after effects of the SMOTE technique shows the change in class distribution before and after data balancing [38]. Splitting the data separates the dataset into subsets for training and testing, ensuring the validity and generalizability of the model. The performance comparison between CNN and LSTM helps in selecting the model that best fits the data and analysis objectives. The confusion matrix provides an overview of the performance of the classification model by displaying the number of correct and incorrect predictions [39], [40]. To evaluate the performance of the model used in this study, several evaluation metrics commonly employed in machine learning and data analysis will be applied. The following are the steps to perform the evaluation [41], [42]:

a. Accuracy: Measures the percentage of correct predictions out of the total predictions. It provides an overview of how well the model works.

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Sample} \tag{1}$$

b. Precision: Measures the accuracy of positive predictions from the model. This is important in cases where the cost of false positives is high.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2}$$

c. Recall (Sensitivity): Measures the ability of the model to detect true positive samples.

$$Recall = \frac{True\ Positive}{True\ Positve + False\ Negative} \tag{3}$$

d. F1-Score: Combines precision and recall into one metric to give a balanced picture of model performance.

$$F1\ Score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recal} \tag{4}$$

III. RESULT AND DISCUSSION

The Deep Learning method is used with the Python programming language and utilizes pre-trained fastText word embeddings for feature extraction. This research is conducted using Google Colab, a cloud-based platform that allows users to write and execute Python code directly in the

INTENSIF, Vol.8 No.2 August 2024
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

browser. Google Colab provides free access to GPUs, which is highly beneficial for accelerating the training process of deep learning models. Additionally, Colab supports various libraries and frameworks needed for this project, such as TensorFlow and PyTorch, and facilitates collaboration by enabling notebook sharing with colleagues. To ensure the success of the proposed model's performance, the evaluation procedure includes dataset training, model evaluation, and scoring metrics.

A. Dataset Labeling

From datasets that have been labeled positive and negative manually look not comparable. so smote is done to resample so that the labeled datasets are comparable. Figure 11 shows the results before resampling and after resampling using SMOTE.

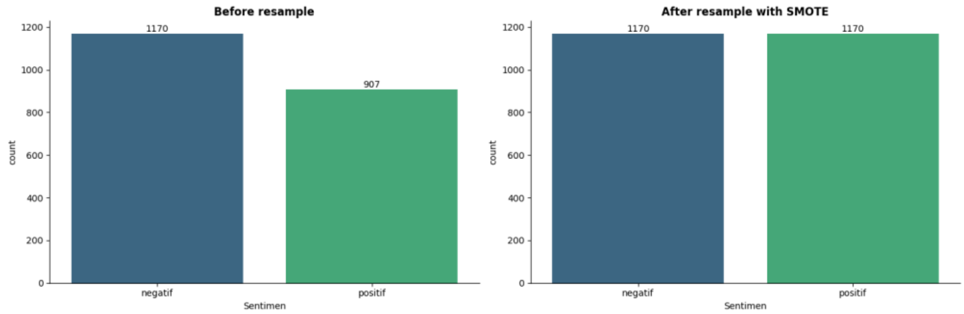

Fig 11. Before Resample and After Resample with Smote

After SMOTE, data splitting is carried out with values of 60%, 70%, 80% and 90%. table 2 below shows the amount of training data that has been split.

Table 2. The Amount of Training Data That Has Been Split

Training Dataset	Label		Total
	Positive	Negative	
60%	702	702	1.404
70%	819	819	1.638
80%	936	936	1.872
90%	1.053	1.053	2106

B. Evaluation Model

Setting the hyperparameter settings for the units, batch size, dropout, and optimizer method is the first step in the model testing process. Tables 3 and 4 display the final hyperparameter testing results for the CNN Architecture and Bidirectional LSTM.

Table 3. Hyperparameter of Architecture CNN

Parameter	Score
Unit	16
Batch Size	64
Dropout	0.5
Optimize	Nadam

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

Table 4. Hyperparameter of Architecture LSTM

Parameter	Score
Unit	16
Batch Size	64
Dropout	0.5
Optimize	Nadam

To find out the performance the model achieved during training, tests on the test size are also conducted. Tables 5 and 6 show the test sizes that were successively tested at 10%, 20%, 30%, and 40% on each architecture.

Table 5. CNN Architecture Test Size

Test Size	Accuracy	Precision	Recall	F-1 Score
0.1	0.8590	0.8590	0.8590	0.8588
0.2	0.8291	0.8291	0.8291	0.8352
0.3	0.8148	0.8148	0.8148	0.8149
0.4	0.7927	0.7927	0.7927	0.7935

Table 6. LSTM Architecture Test Size

Test Size	Accuracy	Precision	Recall	F-1 Score
0.1	0.7991	0.7991	0.7991	0.7906
0.2	0.7821	0.7821	0.7821	0.7837
0.3	0.7450	0.7450	0.7450	0.7451
0.4	0.7714	0.7714	0.7714	0.7708

Accuracy, precision, recall, and F1-score outcomes are compared between CNN and Bidirectional LSTM Architecture. Figures 12 and 13 depict the models' respective performances, which were evaluated using the Confusion Matrix.

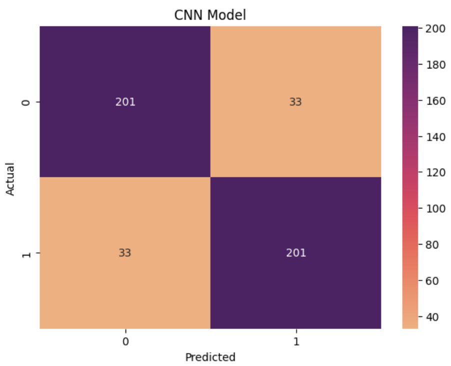

Fig 12. Confusion Matrix CNN

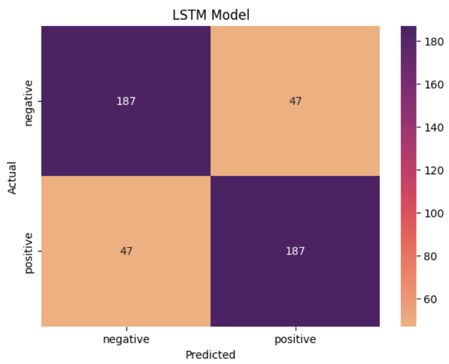

Fig 13. Confusion Matrix LSTM

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

Table 7 demonstrates that the CNN Architecture outperformed the LSTM Architecture, which achieved values of 79.91% for accuracy, 79.91% for precision, 79.91% for recall, and 79.06% for F1-score. The CNN Architecture achieved values of 85.90% for accuracy, 85.90% for precision, 85.90% for recall, and 85.88% for F1-score.

Table 7. Comparison between CNN and LSTM with 10% data test

Algorithm	Accuracy	Precision	Recall	F-1 Score
CNN	0.8590	0.8590	0.8590	0.8588
LSTM	0.7991	0.7991	0.7991	0.7906

CNN architecture performs better than LSTM architecture in training time, averaging 34.76/sec, and 153.18/sec. In order to predict from the past to the future and from the future to the past, LSTM has two LSTM layers. As a result, during the training process, the Bidirectional LSTM architecture requires a longer training time 153.18/sec than CNN, which only requires 34.76/sec. Combining 4019 tokens with Word2Vec 100 dimensions that have already been trained yields excellent results in Indonesian. Table 8 showing for comparasion of CNN & LSTM

Table 8. Comparison accuracy and training time of CNN & LSTM

Architecture	Accuracy	Training Time/sec
CNN	0.8590	34.76
LSTM	0.7991	153.18

IV. CONCLUSION

This research explores the use of deep learning models, specifically CNN and CNN-LSTM, for sentiment analysis on Twitter data concerning Indonesia's Sirekap system. The study highlights the importance of data preprocessing, including steps like cleansing, case folding, tokenizing, and stemming, to prepare the text data for analysis. The findings reveal that CNN architecture outperforms the CNN-LSTM model in terms of accuracy, precision, recall, and F1-score. CNN achieved an accuracy of 85.90%, while CNN-LSTM reached 79.91%. The research also emphasizes the significance of handling class imbalance through techniques like SMOTE to improve model performance. Furthermore, the study demonstrates the effectiveness of the deep learning approach in sentiment analysis, supported by the use of pre-trained word embeddings and the evaluation metrics used. The CNN model's superior performance and faster training time make it a viable choice for future sentiment analysis tasks.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

In comparing the CNN and CNN-LSTM models, the results show a significant difference in performance, with CNN-LSTM generally being expected to capture sequential dependencies better due to its LSTM component. However, in this study, CNN outperformed CNN-LSTM across various metrics. This suggests that while LSTM's capability to retain temporal information is theoretically advantageous, the CNN model's architecture and training efficiency make it better suited for this specific dataset and task.

Additionally, the use of SMOTE (Synthetic Minority Over-sampling Technique) proved essential in addressing class imbalance, positively impacting model performance. With SMOTE, the model was better able to classify minority classes, leading to more balanced predictions. Without SMOTE, the model tended to bias towards the majority class, resulting in suboptimal performance, particularly in minority class classification. Therefore, applying SMOTE is crucial for improving model accuracy in sentiment analysis tasks involving imbalanced data.

Overall, the research provides valuable insights into the application of deep learning models for sentiment analysis and the steps involved in preprocessing and evaluating textual data. The study underscores the potential of these models to enhance the accuracy and efficiency of sentiment classification, contributing to the broader field of natural language processing.

Author Contributions: *Handoko*: Conceptualization, Methodology, Data Curation, Writing - Original Draft, Writing - Review & Editing. *Ahmad Asrofiq*: Software, Investigation, Data Curation. *Junadhi*: Investigation, Supervision, Data Curation. *Ari Sukma Negara*: Software, Investigation, Data Curation

All authors have read and agreed to the published version of the manuscript.

Funding: This research received no specific grant from any funding agency.

Conflicts of Interest: The authors declare no conflict of interest.

Data Availability: Data is available upon request to the corresponding author.

Informed Consent: There were no human subjects.

Animal Subjects: There were no animal subjects.

ORCID:
Handoko: https://orcid.org/0009-0002-6791-7756
Ahmad Asrofiq: https://orcid.org/0009-0003-2520-4472
Junadhi: https://orcid.org/0000-0001-7186-7486
Arik Sukma Negara: https://orcid.org/0009-0001-1205-7421

INTENSIF, Vol.8 No.2 August 2024
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

REFERENCES

[1] "KPU - PAGE." Accessed: Mar. 07, 2024. [Online]. Available: https://www.kpu.go.id/page/read/12/pemilu-dalam-sejarah

[2] "KPU - PAGE." Accessed: Mar. 10, 2024. [Online]. Available: https://www.kpu.go.id/page/read/5/tugas-dan-kewenangan

[3] "Manfaatkan Sirekap, Transparan dan Kemudahan untuk Masyarakat - KPU." Accessed: Mar. 10, 2024. [Online]. Available: https://www.kpu.go.id/berita/baca/10143/manfaatkan-sirekap-transparan-dan-kemudahan-untuk-masyarakat

[4] M. Rodríguez-Ibánez, A. Casánez-Ventura, F. Castejón-Mateos, and P. M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms," *Expert Syst. Appl.*, vol. 223, no. March, 2023, **doi:** 10.1016/j.eswa.2023.119862.

[5] M. K. Anam, M. I. Mahendra, W. Agustin, R. Rahmaddeni, and N. Nurjayadi, "Framework for Analyzing Netizen Opinions on BPJS Using Sentiment Analysis and Social Network Analysis (SNA)," *INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf.*, vol. 6, no. 1, pp. 11–28, 2022, **doi:** 10.29407/intensif.v6i1.15870.

[6] H. T. Phan, V. C. Tran, N. T. Nguyen, and D. Hwang, "Improving the Performance of Sentiment Analysis of Tweets Containing Fuzzy Sentiment Using the Feature Ensemble Model," *IEEE Access*, vol. 8, pp. 14630–14641, 2020, **doi:** 10.1109/ACCESS.2019.2963702.

[7] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4335–4385, 2020, **doi:** 10.1007/s10462-019-09794-5.

[8] Y. Cheng *et al.*, "Sentiment Analysis Using Multi-Head Attention Capsules with Multi-Channel CNN and Bidirectional GRU," *IEEE Access*, vol. 9, pp. 60383–60395, 2021, **doi:** 10.1109/ACCESS.2021.3073988.

[9] M. A. Kausar, A. Soosaimanickam, and M. Nasar, "Public Sentiment Analysis on Twitter Data during COVID-19 Outbreak," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 2, pp. 415–422, 2021, **doi:** 10.14569/IJACSA.2021.0120252.

[10] P. O. A. Sunarya, R. Refianti, A. B. Mutiara, and W. Octaviani, "Comparison of accuracy between convolutional neural networks and Naïve Bayes Classifiers in sentiment analysis on Twitter," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 5, pp. 77–86, 2019, **doi:** 10.14569/ijacsa.2019.0100511.

[11] G. A. M. Kukuh Jaluwana, Gusti Made Arya Sasmita, and I Made Agus Dwi Suarjaya,

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

"Analysis of Public Sentiment Towards Goverment Efforts to Break the Chain of Covid-19 Transmission in Indonesia Using CNN and Bidirectional LSTM," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 4, pp. 511–520, 2022, **doi:** 10.29207/resti.v6i4.4055.

[12] M. Dehghani and Z. Yazdanparast, "Political Sentiment Analysis of Persian Tweets Using CNN-LSTM Model," 2023, **doi:** 10.48550/arXiv.2307.07740.

[13] M. Daffa Dhiyaulhaq, S. Jurnalis Pipin, F. Mikael Sinaga, S. Winardi, and M. Noor Hakim, "Sentiment Analysis Classification of ChatGPT on Twitter Big Data in Indonesia Using Fast R-CNN," *Media Inform. Budidarma*, vol. 7, no. 4, pp. 2137–2148, 2023, **doi:** 10.30865/mib.v7i4.6816.

[14] H. Yanagimoto and S. Isaji, "Incident related tweet extraction with density ratio estimation," *Procedia Comput. Sci.*, vol. 35, no. C, pp. 456–463, 2014, **doi:** 10.1016/j.procs.2014.08.126.

[15] R. Wongso, F. A. Luwinda, B. C. Trisnajaya, O. Rusli, and Rudy, "News Article Text Classification in Indonesian Language," *Procedia Comput. Sci.*, vol. 116, pp. 137–143, 2017, **doi:** 10.1016/j.procs.2017.10.039.

[16] M. Rodriguez-Ibanez, F. J. Gimeno-Blanes, P. M. Cuenca-Jimenez, C. Soguero-Ruiz, and J. L. Rojo-Alvarez, "Sentiment Analysis of Political Tweets from the 2019 Spanish Elections," *IEEE Access*, vol. 9, pp. 101847–101862, 2021, **doi:** 10.1109/ACCESS.2021.3097492.

[17] V. A. Fitri, R. Andreswari, and M. A. Hasibuan, "Sentiment analysis of social media Twitter with case of Anti-LGBT campaign in Indonesia using Naïve Bayes, decision tree, and random forest algorithm," *Procedia Comput. Sci.*, vol. 161, pp. 765–772, 2019, **doi:** 10.1016/j.procs.2019.11.181.

[18] M. J. Tan and C. H. Guan, "Are people happier in locations of high property value? Spatial temporal analytics of activity frequency, public sentiment and housing price using twitter data," *Appl. Geogr.*, vol. 132, p. 102474, 2021, **doi:** 10.1016/j.apgeog.2021.102474.

[19] B. Kanwal *et al.*, "Opinion Mining from Online Travel Reviews: An Exploratory Investigation on Pakistan Major Online Travel Services Using Natural Language Processing," *IEEE Access*, vol. 11, no. March, pp. 29934–29945, 2023, **doi:** 10.1109/ACCESS.2023.3260114.

[20] A. G. Gozal, H. Pranoto, and M. F. Hasani, "Sentiment analysis of the Indonesian community toward face-to-face learning during the Covid-19 pandemic," *Procedia Comput. Sci.*, vol. 227, pp. 398–405, 2023, **doi:** 10.1016/j.procs.2023.10.539.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

[21] F. Zuhad and N. Wilantika, "Perbandingan Penggunaan Kamus Normalisasi dalam Analisis Sentimen Berbahasa Indonesia," *J. Linguist. Komputasional*, vol. 5, no. 1, pp. 13–23, 2022, **doi:** 10.26418/jlk.v5i1.60.

[22] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The Impact of Features Extraction on the Sentiment Analysis," *Procedia Comput. Sci.*, vol. 152, pp. 341–348, Jan. 2019, **doi:** 10.1016/J.PROCS.2019.05.008.

[23] M. Demircan, A. Seller, F. Abut, and M. F. Akay, "Developing Turkish sentiment analysis models using machine learning and e-commerce data," *Int. J. Cogn. Comput. Eng.*, vol. 2, no. November, pp. 202–207, 2021, **doi:** 10.1016/j.ijcce.2021.11.003.

[24] A. Erkan and T. Gungor, "Analysis of Deep Learning Model Combinations and Tokenization Approaches in Sentiment Classification," *IEEE Access*, vol. 11, no. December, pp. 134951–134968, 2023, **doi:** 10.1109/ACCESS.2023.3337354.

[25] D. Ramachandran and R. Parvathi, "Analysis of Twitter Specific Preprocessing Technique for Tweets," *Procedia Comput. Sci.*, vol. 165, pp. 245–251, 2019, **doi:** 10.1016/j.procs.2020.01.083.

[26] M. Young *et al.*, "Natural language processing to assess the epidemiology of delirium-suggestive behavioural disturbances in critically ill patients," *Crit. Care Resusc.*, vol. 23, no. 2, pp. 144–153, 2021, **doi:** 10.51893/2021.2.oa1.

[27] A. Jabbar, S. Iqbal, M. I. Tamimy, A. Rehman, S. A. Bahaj, and T. Saba, "An Analytical Analysis of Text Stemming Methodologies in Information Retrieval and Natural Language Processing Systems," *IEEE Access*, vol. 11, no. December, pp. 133681–133702, 2023, **doi:** 10.1109/ACCESS.2023.3332710.

[28] "Sastrawi · PyPI." Accessed: Aug. 28, 2024. [Online]. Available: https://pypi.org/project/Sastrawi/

[29] H. S. Hota, D. K. Sharma, and N. Verma, *Lexicon-based sentiment analysis using Twitter data*. Elsevier Inc., 2021. **doi:** 10.1016/B978-0-12-824536-1.00015-0.

[30] C. Padurariu and M. E. Breaban, "Dealing with data imbalance in text classification," *Procedia Comput. Sci.*, vol. 159, pp. 736–745, 2019, **doi:** 10.1016/j.procs.2019.09.229.

[31] V. Rupapara, F. Rustam, H. F. Shahzad, A. Mehmood, I. Ashraf, and G. S. Choi, "Impact of SMOTE on Imbalanced Text Features for Toxic Comments Classification Using RVVC Model," *IEEE Access*, vol. 9, pp. 78621–78634, 2021, **doi:** 10.1109/ACCESS.2021.3083638.

[32] A. Arafa, N. El-Fishawy, M. Badawy, and M. Radad, "RN-SMOTE: Reduced Noise SMOTE based on DBSCAN for enhancing imbalanced data classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5059–5074, 2022, **doi:**

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.23046

10.1016/j.jksuci.2022.06.005.

[33] "Splitting Datasets With scikit-learn and train_test_split() (Overview) – Real Python." Accessed: May 25, 2024. [Online]. Available: https://realpython.com/lessons/splitting-datasets-overview/

[34] S. Tammina and S. Annareddy, "Sentiment analysis on customer reviews using convolutional neural network," *2020 Int. Conf. Comput. Commun. Informatics, ICCCI 2020*, pp. 20–25, 2020, **doi:** 10.1109/ICCCI48352.2020.9104086.

[35] S. Tam, R. Ben Said, and Ö. Tanriöver, "A ConvBiLSTM Deep Learning Model-Based Approach for Twitter Sentiment Classification," *IEEE Access*, vol. 9, pp. 41283–41293, 2021, **doi:** 10.1109/ACCESS.2021.3064830.

[36] W. Meng, Y. Wei, P. Liu, Z. Zhu, and H. Yin, "Aspect Based Sentiment Analysis with Feature Enhanced Attention CNN-BiLSTM," *IEEE Access*, vol. 7, pp. 167240–167249, 2019, **doi:** 10.1109/ACCESS.2019.2952888.

[37] J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Tree-Structured Regional CNN-LSTM Model for Dimensional Sentiment Analysis," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 581–591, 2020, **doi:** 10.1109/TASLP.2019.2959251.

[38] Y. Bao and S. Yang, "Two Novel SMOTE Methods for Solving Imbalanced Classification Problems," *IEEE Access*, vol. 11, no. December 2022, pp. 5816–5823, 2023, **doi:** 10.1109/ACCESS.2023.3236794.

[39] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: Multi-Label Confusion Matrix," *IEEE Access*, vol. 10, pp. 19083–19095, 2022, **doi:** 10.1109/ACCESS.2022.3151048.

[40] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020, **doi:** 10.1186/s12864-019-6413-7.

[41] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning," *IEEE Access*, vol. 8, pp. 23522–23530, 2020, **doi:** 10.1109/ACCESS.2020.2969854.

[42] Ç. Oğuz and M. Yağanoğlu, "Detection of COVID-19 using deep learning techniques and classification methods," *Inf. Process. Manag.*, vol. 59, no. 5, pp. 1–18, 2022, **doi:** 10.1016/j.ipm.2022.103025.