ISSN : 2527-5836
e-ISSN : 2528-0074

Vol. 10 No. 1, January 2025

JISKa

Jurnal Informatika Sunan Kalijaga

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta

JISKa Editorial Team

Januari 2025 Edition

Reviewers

Agung Dewandaru, Institut Teknologi Bandung, Indonesia

Agus Mulyanto, UIN Sunan Kalijaga Yogyakarta, Indonesia

Ahmad Fathan Hidayatullah, Universitas Islam Indonesia Yogyakarta, Indonesia

Alam Rahmatulloh, Universitas Siliwangi Tasikmalaya, Indonesia

Anggi Rizky Windra Putri, Universitas Aisyiyah Yogyakarta, Indonesia

Ardiansyah Musa Efendi, Singapore Chipset Algorithm Design Lab, Huawei, Singapore

Bambang Sugiantoro, UIN Sunan Kalijaga Yogyakarta, Indonesia

Enny Itje Sela, Universitas Teknologi Yogyakarta, Indonesia

Ganjar Alfian, Universitas Gadjah Mada, Indonesia

Mandahadi Kusuma, UIN Sunan Kalijaga Yogyakarta, Indonesia

Maria Ulfah Siregar, UIN Sunan Kalijaga Yogyakarta, Indonesia

Millati Pratiwi, Pusan National University, South Korea

Muhammad Dzulfikar Fauzi, Telkom University Surabaya, Indonesia

Muhammad Habibi, Universitas Jenderal Achmad Yani Yogyakarta, Indonesia

Muhammad Rifqi Maarif, Universitas Jenderal Achmad Yani Yogyakarta, Indonesia

Mohd. Fikri Azli bin Abdullah, Multimedia University, Malaysia

M. Alex Syaekhoni, Who's Good, South Korea

Niki Min Hidayati Robbi, Universitas Gadjah Mada, Indonesia

Norma Latif Fitriyani, Sejong University Seoul, South Korea

Okfalisa, UIN Sultan Syarif Kasim Riau, Indonesia

Oman Somantri, Politeknik Negeri Cilacap, Indonesia

Puguh Jayadi, Universitas PGRI Madiun, Indonesia

Puji Winar Cahyo, Universitas Jenderal Achmad Yani Yogyakarta, Indonesia

Qorry Aina Fitroh, UIN KH. Abdurrahman Wahid Pekalongan, Indonesia

Ridho Surya Kusuma, Universitas Siber Muhammadiyah, Yogyakarta, Indonesia

Rischan Mafrur, Macquarie University, Sydney, Australia

Shofwatul Uyun, UIN Sunan Kalijaga Yogyakarta, Indonesia

Sumarsono, UIN Sunan Kalijaga, Indonesia

Sunu Wibirama, Universitas Gadjah Mada, Indonesia

Tundo, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI), Indonesia

Windra Swastika, Universitas Ma Chung, Indonesia

Yudistira Dwi Wardhana Asnar, Institut Teknologi Bandung, Indonesia

ISSN : 2527-5836

e-ISSN: 2528-0074

JISKa (Jurnal Informatika Sunan Kalijaga)

Vol. 10, No. 1, JANUARY 2025

TABLE OF CONTENT

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 1 – 15
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Optimizing K-Means Algorithm Using the Purity Method for Clustering Oil Palm Producing Regions

Novia Hasdyna [(1)*], Rozzi Kesuma Dinata [(2)], Balqis Yafis [(3)]
[1] Department of Informatics, Universitas Islam Kebangsaan Indonesia, Bireuen, Indonesia
[2] Department of Informatics Engineering, Universitas Malikussaleh, Aceh Utara, Indonesia
[3] Department of Electronics and Electrical Engineering, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
e-mail : noviahasdyna@uniki.ac.id, rozzi@unimal.ac.id, balqisyafis@nycu.edu.tw.
* Corresponding author.
This article was submitted on 15 October 2024, revised on 2 November 2024, accepted on 3 November 2024, and published on 31 January 2025.

Abstract

The K-Means algorithm is a fundamental tool in machine learning, widely utilized for data clustering tasks. This research aims to improve the performance of the K-Means algorithm by integrating the Purity method, specifically focusing on clustering regions renowned for oil palm production in North Aceh. Oil palm cultivation is a vital agricultural sector in North Aceh, contributing significantly to the local economy and employment. This study examines two clustering techniques: the conventional K-Means algorithm and an optimized version, Purity K-Means. Integrating the Purity method increases K-Means' efficiency by decreasing the required convergence iteration. The data used for clustering analysis is sourced from the Department of Agriculture and Food in North Aceh Regency and pertains to oil palm production in 2023. The findings indicate that the Purity K-Means approach notably reduces the iteration count and improves cluster quality. The average Davies-Bouldin Index (DBI) for standard K-Means is 0.45, whereas the Purity K-Means method lowers it to 0.30. Furthermore, applying the Purity method reduced the number of K-Means iterations from 15 to just 3. These results highlight an enhancement in clustering performance and overall efficiency.

Keywords: K-Means Algorithm, Purity Method, Data Clustering, Oil Palm Production, Davies-Bouldin Index (DBI)

Abstrak

Algoritma K-Means merupakan alat dasar dalam pembelajaran mesin yang banyak digunakan untuk tugas pengelompokan data. Penelitian ini bertujuan untuk meningkatkan kinerja algoritma K-Means dengan mengintegrasikan metode Purity, yang secara khusus difokuskan pada pengelompokan wilayah-wilayah yang terkenal dengan produksi kelapa sawit di Aceh Utara. Budidaya kelapa sawit merupakan sektor pertanian yang vital di Aceh Utara, memberikan kontribusi signifikan terhadap perekonomian lokal dan penyerapan tenaga kerja. Studi ini membandingkan dua pendekatan pengelompokan, yaitu K-Means standar dan Purity K-Means yang telah dioptimalkan. Metode Purity digunakan untuk meningkatkan efisiensi algoritma K-Means dengan mengurangi jumlah iterasi yang diperlukan untuk konvergensi. Data yang digunakan dalam analisis pengelompokan bersumber dari Dinas Pertanian dan Pangan Kabupaten Aceh Utara dan berkaitan dengan produksi kelapa sawit pada tahun 2023. Hasil penelitian menunjukkan bahwa pendekatan Purity K-Means secara signifikan mengurangi jumlah iterasi dan meningkatkan kualitas cluster. Nilai rata-rata Davies-Bouldin Index (DBI) untuk K-Means standar adalah 0,45, sedangkan metode Purity K-Means menguranginya menjadi 0,30. Selain itu, jumlah iterasi K-Means berkurang dari 15 menjadi 3 saat menggunakan metode Purity. Temuan ini mengindikasikan peningkatan kinerja pengelompokan dan efisiensi secara keseluruhan.

Kata Kunci: Algoritma K-Means, Metode Purity, Pengelompokan Data, Produksi Kelapa Sawit, Davies-Bouldin Index (DBI)

1. INTRODUCTION

Clustering is an essential data analysis technique that groups similar objects based on specific attributes, providing valuable insights across various applications (Ezugwu et al., 2022). Among the many clustering algorithms available, the K-Means algorithm is particularly notable for its widespread use due to its simplicity, efficiency, and effectiveness in handling large datasets (Kouadio et al., 2024; Li et al., 2023). However, K-Means has its limitations; one significant challenge is the high number of iterations required for convergence, which can increase computational time and overall processing demands, especially with large datasets (Cebolla-Alemany et al., 2024). This study seeks to address these limitations by integrating the Purity method to enhance the efficiency of the K-Means algorithm.

The focus on North Aceh's oil palm production is grounded in the region's economic reliance on this sector, which plays a key role in local employment and economic growth. As demand for oil palm products increases, analyzing and understanding production data in regions like North Aceh has become essential. However, the large agricultural datasets generated in this sector pose significant challenges for traditional clustering methods, especially regarding scalability and meaningful data grouping. Therefore, this research aims to contribute to understanding and analyzing oil palm production in North Aceh. In this region, data-driven insights can substantially impact local and regional development.

In recent years, advances in clustering algorithms have shown significant potential for agricultural applications, where techniques such as hierarchical clustering, K-Medoids, and others have been employed to manage and analyze agricultural data effectively and efficiently. The Purity method, in particular, provides a valuable approach by enhancing cluster homogeneity, thus improving cluster interpretability and consistency. By integrating Purity with K-Means, this study aims to improve clustering quality and reduce the number of iterations, enabling more efficient processing of complex agricultural datasets.

A literature review shows various studies that have explored the application of clustering algorithms in agricultural contexts. For example, Majumdar et al. (2023) used K-Means to optimize irrigation management in rice production, leading to better yield predictions. Similarly, Rezaee et al. (2023) applied K-Means to classify soil types based on diverse attributes, highlighting their effectiveness in agricultural land management. Naz et al. (2024) utilized K-Means to analyze crop yield data, identifying patterns that improved resource allocation. Thakur & Kaur (2024) used K-Means to identify potential areas for organic farming, demonstrating its value in promoting sustainable practices. Finally, Bhatti et al. (2024) combined K-Means with other machine learning techniques to improve crop disease prediction, showing the algorithm's adaptability in various agricultural applications. Despite these advancements, limited focus has been on optimizing K-Means using methods like Purity. This study seeks to address this gap by examining the potential of the Purity method to enhance K-Means performance, particularly for oil palm production data.

The objectives of this study are as follows:

a) To optimize the K-Means algorithm by integrating the Purity method, specifically focusing on clustering regions known for oil palm production in North Aceh.
b) To evaluate the performance of the standard K-Means algorithm compared to the optimized Purity K-Means approach, using data from the Department of Agriculture and Food in North Aceh Regency from 2023.
c) The effects of the purity method on the number of iterations required for convergence and overall clustering quality will be analyzed using the Davies-Bouldin Index (DBI).

This study hypothesizes that integrating the Purity method with the K-Means algorithm will significantly reduce the number of iterations required for convergence and enhance clustering quality, as indicated by a lower Davies-Bouldin Index (DBI) compared to the standard K-Means approach.

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025: 1 – 15

2. METHODS

This research adopts a quantitative approach to analyze the clustering of oil palm production regions in North Aceh. The study compares two clustering methodologies: the conventional K-Means algorithm and the enhanced Purity K-Means algorithm. The research design is structured to facilitate the assessment of the performance of these methods using agricultural data.

2.1 Dataset Preparation

The dataset utilized for this research consists of data related to oil palm production in North Aceh for 2023, as presented in Table 1. This data was sourced from the Department of Agriculture and Food in North Aceh Regency. It encompasses several key features that are essential for analyzing the factors influencing oil palm production, including:

a) Production Volume (X1): This feature represents the total volume of oil palm produced in each region, quantified in metric tons.
b) Land Area (X2): This denotes the area allocated for oil palm cultivation, measured in hectares.
c) Yield per Hectare (X3): This variable reflects the average oil palm yield per hectare, offering valuable insights into agricultural productivity.

Table 1 Research Dataset

No.	District Name	Production Volume	Land Area	Yield per Hectare
1	Sawang	0,851	11,388	15,600
2	Nisam	0,727	10,189	15,700
3	Nisam Antara	0,465	6,726	16,900
4	Kuta Makmur	2,388	39,603	17,500
5	Syamtalira Bayu	0,454	7,012	15,900
6	Geureudong Pase	0,952	13,675	15,540
7	Samudera	0,018	0,252	14,000
8	Meurah Mulia	0,461	5,277	15,800
9	Tanah Luas	0,441	5,379	16,500
10	Matang Kuli	0,358	1,766	16,500
11	Pirak Timu	0,380	4,051	16,400
12	Lhoksukon	2,170	35,055	16,520
13	Baktiya	1,047	16,286	16,500
14	Tanah Jambo Aye	1,629	18,431	16,500
15	Cot Girek	2,597	40,340	17,188
16	Langkahan	2,188	34,122	16,500
17	Baktiya Barat	0,100	1,504	15,500
18	Paya Bakong	0,423	3,185	16,500
19	Nibong	0,043	0,375	15,000
20	Simpang Kramat	0,410	4,603	16,800

Table 1 outlines the dataset utilized in this study, encompassing data from 20 districts in North Aceh related to oil palm production for 2023. Each entry provides distinct characteristics for the districts, offering a snapshot of agricultural dynamics in the region. The dataset is vital for evaluating the varying oil palm output levels and understanding the land distribution dedicated to this crucial crop. By analyzing these parameters, researchers can derive insights into regional agricultural practices and identify potential areas for improvement and intervention. This diverse dataset facilitates a comprehensive examination of factors that may influence oil palm production across different districts in North Aceh.

2.2 Proposed Model

This research introduces two distinct models for clustering oil palm-producing regions in North Aceh: the Purity K-Means model and the conventional K-Means model. Both models aim to

enhance the clustering process but differ in their methodologies and implementation. The Purity K-Means model integrates the standard K-Means algorithm with the Purity method to optimize the clustering process, as shown in Figure 1.

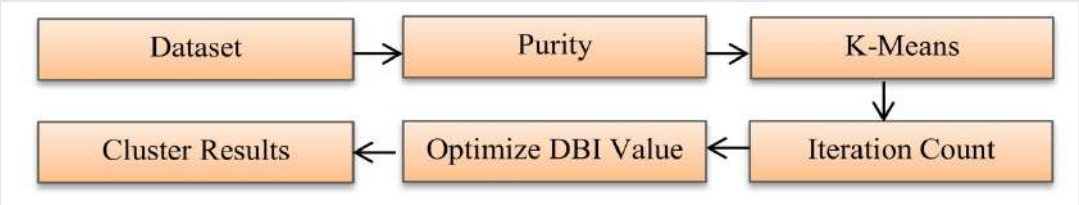


Figure 1 Purity K-Means Model

In Figure 1, the stages of the Purity K-Means process are as follows. For the data preparation, the dataset is normalized to ensure equal contribution from each feature during the clustering process. This step is essential for minimizing bias associated with varying feature scales. For purity calculation, after each iteration, the model calculates the Purity score for the generated clusters. This score evaluates cluster homogeneity, providing insights into the effectiveness of centroid adjustments and data point assignments. Purity is utilized to assess a cluster's purity value, identifying the most suitable cluster member within a class (Dinata et al., 2023). The formula for calculating Purity is presented in Equation (1). Purity (y) reflects the purity level for the y-variable, where N_y represents the total data points within the y-cluster, and y signifies the cluster index (Hasdyna & Dinata, 2024).

$$Purity\ (y) = \frac{1}{N_y} \max\ (n_{xy}) \tag{1}$$

Instead of random initialization, this model uses the Purity method to select initial centroids. This approach identifies centroids representing the data's inherent structure, improving the chances of forming meaningful clusters from the outset. For clustering process using k-means, the algorithm iteratively assigns data points to the nearest centroid, recalculates centroids based on current assignments, and repeats this process until convergence. The integration of the Purity method allows for continuous assessment of cluster quality during iterations. The K-means algorithm is applied for data clustering. The clustering process with K-means follows these steps: In Step 1, the desired number of clusters, denoted by 'k,' is determined. In Step 2, initial random values are assigned to the centroids of each of the 'k' clusters. The Euclidean distance formula is then used to calculate the distance between each data point and the centroids, shown in Equation (2) (Ariyanto et al., 2024).

$$d(xi, \mu j) = \sqrt{\sum (xi - \mu j)^2} \tag{2}$$

Here, d represents a data point, xi denotes the data criteria, and μj indicates the cluster j's centroid. In Step 3, each data point is assigned to the cluster of the nearest centroid. Step 4 involves updating the centroids by calculating the mean of the data points within each cluster using the formula in Equation (3) (Retno et al., 2024).

$$\mu j(t+1) = \frac{1}{Nsj} \sum_{j \in sj} xj \tag{3}$$

In this context, the symbol $\mu j(t+1)$ represents the centroid updated at iteration $t+1$, indicating the evolving center of a specific cluster. The term Nsj corresponds to the dataset contained within the Sj cluster, signifying the collection of data points grouped in that particular cluster. Additionally, xj represents the cumulative values within cluster Sj, effectively summarizing the overall attributes of the clustered data points. Finally, Step 5 marks the completion of the process. Steps 2 through

4 are repeated until no further changes occur in cluster membership, confirming convergence and consistent cluster assignments.

The model's performance is evaluated using metrics such as the Davies-Bouldin Index (DBI) and the number of iterations needed for convergence. These metrics facilitate clustering quality and efficiency assessment with the traditional K-Means method. The initial step in calculating the DBI involves determining the Sum of Squares Within the Cluster (SSW), which indicates the cohesion value. The DBI is then computed using the formula presented in Equation (4) (Ros et al., 2023).

$$SSW_i = \frac{1}{mi}\sum_{j=i}^{mi} d(xj, ci) \tag{4}$$

Once SSW has been computed, the subsequent step involves calculating the Sum of Squares Between Clusters (SSB), which reflects the cluster separation value. This is achieved using the formula presented in Equation (5) (Henderi et al., 2024).

$$SSB_{i,j} = d(c_i, c_j) \tag{5}$$

The following step involves calculating the Ratio to compare the i-cluster with the j-cluster, utilizing the formula in Equation (6).

$$R_{ij} = \frac{SSW_i + SSW_j}{SSB_{ij}} \tag{6}$$

After deriving the ratio value, the final step is to compute the DBI value using the formula provided in Equation (7).

$$DBI = \frac{1}{k}\sum_{i=1}^{k} max_{i \neq j}(R_{i,j}) \tag{7}$$

In the context of DBI, a smaller value signifies better clustering results, indicating that the clusters are more internally cohesive and distinct, which is ideal for clustering tasks.

Secondly, the Conventional K-Means model serves as a benchmark for evaluating the effectiveness of the Purity K-Means model. The steps involved in this model are outlined as follows. For data preparation, like the Purity K-Means model, the dataset undergoes normalization to ensure that all features contribute equally to the clustering process. In this model, initial centroids are selected randomly from the dataset. This randomness can lead to varying clustering outcomes across different runs. For the clustering process, the K-Means algorithm iteratively assigns data points to the nearest centroid and recalculates centroids based on the assigned points. This process continues until convergence, which may take multiple iterations. The performance of the conventional K-Means model is measured using the Davies-Bouldin Index (DBI) and the total number of iterations required for convergence. These metrics serve as indicators of clustering quality and efficiency.

3. RESULTS AND DISCUSSION

3.1 Purity calculation results

Table 2 and Figure 2 present the purity calculation results. To initiate K-Means clustering using purity values as centroids, we selected three representative subdistricts based on their purity scores: Samudera (high Purity, 0.9811), Langkahan (medium Purity, 0.6461), and Baktiya (low Purity, 0.4877). Samudera exhibits highly consistent attributes with the highest Purity (X1, X2, X3), making it an ideal centroid for clustering subdistricts with similar stability. Langkahan, with a moderate purity value and a balanced attribute sum of 52.81, serves as a centroid that captures

subdistricts with average consistency. In contrast, Baktiya, having one of the lowest purity scores, indicates a high degree of attribute variability, making it suitable for clustering subdistricts with less consistency or greater diversity in attributes. These centroids will allow us to analyze groupings based on stability and variance within subdistrict attributes.

Table 2 Purity Calculation Results

No.	District Name	Production Volume	Land Area	Yield per Hectare	∑	Purity
1	Sawang	0,851	11,388	15,600	27,839	0,560364956
2	Nisam	0,727	10,189	15,700	26,616	0,589870754
3	Nisam Antara	0,465	6,726	16,900	24,091	0,701506787
4	Kuta Makmur	2,388	39,603	17,500	59,491	0,665697332
5	Syamtalira Bayu	0,454	7,012	15,900	23,366	0,680475905
6	Geureudong Pase	0,952	13,675	15,540	30,167	0,515132429
7	Samudera	0,018	0,252	14,000	14,27	0,981079187
8	Meurah Mulia	0,461	5,277	15,800	21,538	0,733587148
9	Tanah Luas	0,441	5,379	16,500	22,32	0,739247312
10	Matang Kuli	0,358	1,766	16,500	18,624	0,885953608
11	Pirak Timu	0,380	4,051	16,400	20,831	0,787288176
12	Lhoksukon	2,170	35,055	16,520	53,745	0,652246721
13	Baktiya	1,047	16,286	16,500	33,833	0,487689534
14	Tanah Jambo Aye	1,629	18,431	16,500	36,56	0,504130197
15	Cot Girek	2,597	40,340	17,188	60,125	0,670935551
16	Langkahan	2,188	34,122	16,500	52,81	0,646127627
17	Baktiya Barat	0,100	1,504	15,500	17,104	0,906220767
18	Paya Bakong	0,423	3,185	16,500	20,108	0,820568928
19	Nibong	0,043	0,375	15,000	15,418	0,972888831
20	Simpang Kramat	0,410	4,603	16,800	21,813	0,770182918

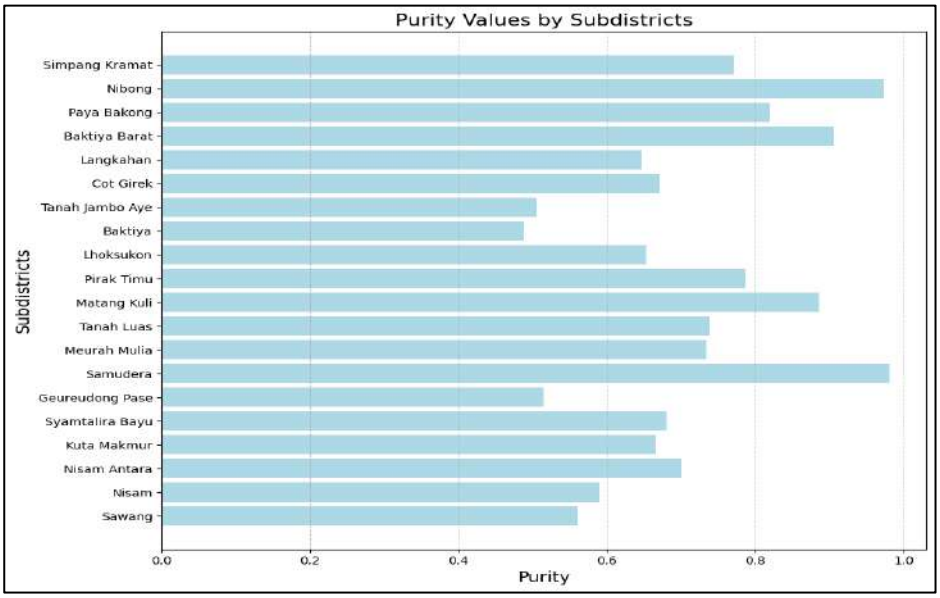


Figure 2 Purity Values in the Oil Palm Dataset

Table 3 outlines the initial centroids selected for K-Means clustering based on purity levels of three subdistricts. Samudera, with the highest purity value of 0.9811, signifies a cluster characterized by highly uniform attribute distributions (X1, X2, X3). This suggests that subdistricts grouped around Samudera are likely to share similar socioeconomic or environmental conditions, making it an effective point for clustering those with stable characteristics. In contrast, Langkahan, exhibiting a medium purity score of 0.6461, reflects a blend of consistency and variability in its attributes. This makes it an appropriate centroid for clustering subdistricts with average attributes, thereby capturing a wider range of subdistrict profiles without leaning too heavily towards extremely high or low consistency.

Table 3 Initial Centroids for K-Means Clustering Based on Purity Levels

Purity Level	Subdistrict	Purity Value	Characteristics
High Purity	Samudera	0.9811	Highly consistent attributes (X1, X2, X3), suitable as a centroid for stable subdistricts
Medium Purity	Langkahan	0.6461	Moderate consistency, with a balanced attribute sum of 52.81, ideal for capturing average subdistricts
Low Purity	Baktiya	0.4877	High variability in attributes, suitable for clustering subdistricts with less consistency or greater diversity

On the other hand, Baktiya, with the lowest purity score of 0.4877, indicates considerable diversity within its attributes. By choosing Baktiya as a low-purity centroid, the clustering process can effectively encompass subdistricts with more pronounced variations in their characteristics. This allows for identifying groups that may experience diverse conditions, which could be critical for targeted interventions or resource allocation. Overall, selecting these three subdistricts as centroids based on their purity scores allows for a nuanced approach in clustering, capturing varying levels of consistency and diversity across the dataset. This stratified methodology is beneficial for understanding the different dynamics present within the region.

3.2 Clustering process using Purity K-Means

To manually do K-Means clustering using the initial centroids provided in Table 4., the following steps will be undertaken. To select initial centroids, the subdistricts Samudera, Langkahan, and Baktiya will be designated as the initial centroids, as shown in Table 4. As for cluster assignment, For each subdistrict, we will compute the Euclidean distance to each centroid and assign the subdistrict to the nearest centroid based on the calculated distances, as shown in Table 5. After assigning the clusters, we will determine the new centroids by calculating the mean attribute values of the subdistricts within each cluster, as shown in Table 6.

Table 4 Initial Centroids for K-Means Clustering

Centroid	Subdistrict	X1	X2	X3
C1	Samudera	0.018	0.252	14.000
C2	Langkahan	2.188	34.122	16.500
C3	Baktiya	1.047	16.286	16.500

The assignment and update steps will be repeated until the centroids converge, when they no longer change significantly, or when the cluster assignments remain constant. The purity K-Means clustering process reached convergence after three iterations, the result shown in Table 7, where the centroids stabilized, indicating that further adjustments in cluster assignments were no longer necessary. In the first iteration, the initial centroids, Samudera, Langkahan, and Baktiya, were assigned to clusters based on the proximity of subdistricts, resulting in new centroid calculations. Subsequent iterations demonstrated a gradual refinement of cluster assignments and centroid positions, reflecting the algorithm's effectiveness in identifying distinct groupings among the subdistricts based on their attributes. Ultimately, the stability achieved in the third

iteration suggests that the clustering solution accurately captures the underlying patterns in the data, enabling meaningful insights into the characteristics of each subdistrict based on their calculated purity values.

Table 5 Euclidean Distances to Initial Centroids

No.	Subdistrict	Distance to C1 (Samudera)	Distance to C2 (Langkahan)	Distance to C3 (Baktiya)
1	Sawang	28.837	29.356	24.098
2	Nisam	27.769	27.054	25.274
3	Nisam Antara	17.172	19.058	16.185
4	Kuta Makmur	41.266	43.942	40.351
5	Syamtalira Bayu	23.097	23.719	22.005
6	Geureudong Pase	29.205	31.187	27.602
7	Samudera	0.000	30.250	14.000
8	Meurah Mulia	21.721	23.169	18.071
9	Tanah Luas	24.640	25.712	22.825
10	Matang Kuli	18.448	20.226	16.070
11	Pirak Timu	20.096	21.798	18.000
12	Lhoksukon	86.424	86.956	53.245
13	Baktiya	31.135	32.530	0.000
14	Tanah Jambo Aye	36.435	36.564	36.198
15	Cot Girek	60.125	61.892	29.382
16	Langkahan	52.610	52.013	52.241
17	Baktiya Barat	17.104	19.073	16.817
18	Paya Bakong	20.108	22.760	19.907
19	Nibong	15.417	16.300	14.628
20	Simpang Kramat	21.813	23.013	20.218

Table 6 Updated Centroids After First Iteration

Centroid	Coordinates (X1, X2, X3)
C1	(0.0305, 0.3135, 14.500)
C2	(2.33575, 37.029, 17.077)
C3	(0.553, 7.928, 16.444)

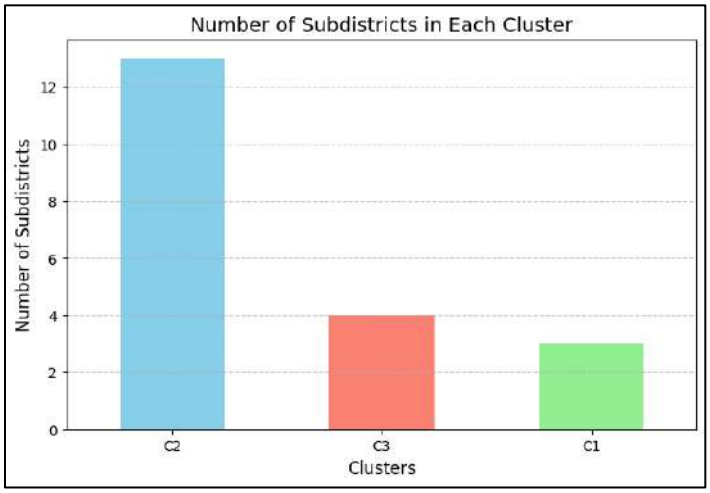


Figure 3 Distribution of Clustering Results with Purity K-Means

In the context of clustering oil palm-producing regions in North Aceh, the Purity K-Means analysis identified three distinct clusters that highlight varying regional characteristics, as shown in Figure 3. Cluster C1, which includes subdistricts like Samudera and Nibong, demonstrates high purity values, suggesting these areas have similar environmental and economic conditions conducive to oil palm production. This consistency may indicate effective agricultural practices or favorable land conditions, warranting the implementation of targeted agricultural policies to enhance production efficiency. Cluster C2 consists of larger subdistricts, such as Langkahan and Cot Girek, showcasing moderate attribute consistency.

Table 7 Purity K-Means Clustering Results for Subdistricts

Subdistricts	Cluster
Samudera	C1
Nibong	C1
Langkahan	C2
Cot Girek	C2
Baktiya	C3
Nisam Antara	C3
Sawang	C1
Nisam	C2
Kuta Makmur	C2
Syamtalira Bayu	C2
Geureudong Pase	C2
Meurah Mulia	C2
Tanah Luas	C2
Matang Kuli	C3
Pirak Timu	C2
Lhoksukon	C2
Tanah Jambo Aye	C3
Langkahan	C2
Paya Bakong	C2
Simpang Kramat	C2

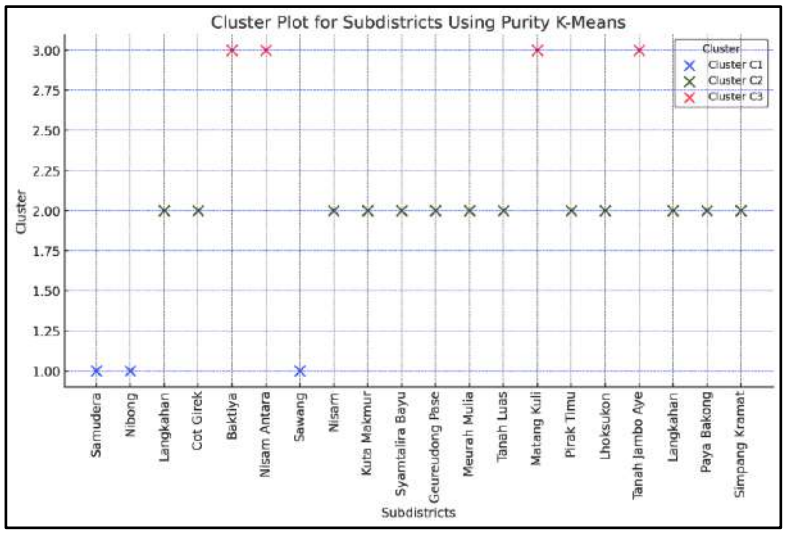


Figure 4 Visualization of the Clustering Results using Conventional K-Means

This cluster likely represents regions with diverse agricultural practices and varying production levels, indicating a need for tailored support and resource allocation to optimize their oil palm outputs. Conversely, Cluster C3, which encompasses Baktiya and Nisam Antara, exhibits lower

purity values, highlighting greater variability in production conditions and practices. The unique challenges faced by these regions may require specialized interventions or research to improve their oil palm production capabilities. Overall, these clustering results provide critical insights that can inform strategic decision-making and development efforts in the oil palm sector of North Aceh. The visualization of the clustering results using Purity K-Means is presented in Figure 4.

3.3 Results of the Conventional K-Means Model

In conventional K-Means, the initial centroids are selected randomly, unlike in Purity K-Means, where the initial centroids are chosen based on purity results. Table 8 provides an overview of the initial centroids assigned randomly for three regions within the conventional K-Means clustering model: Nisam Antara, Samudera, and Pirak Timu. These centroids represent starting points for each cluster, specifically in terms of three selected variables (Value 1, Value 2, and Value 3) relevant to the clustering analysis. The distance calculation results are presented in Table 9.

Table 8 Initial Centroids for K-Means Clustering

Centroid	Subdistrict	X1	X2	X3
C1	Nisam Antara	0,103	1,656	16,900
C2	Samudera	0,018	0,252	14,000
C3	Pirak Timu	0,285	4,051	16,400

Table 9 Euclidean Distances to Initial Centroids in Conventional K-means

No.	Subdistrict	Distance to C1 (Samudera)	Distance to C2 (Langkahan)	Distance to C3 (Baktiya)
1	Sawang	9,847	11,281	7,402
2	Nisam	8,640	10,106	6,194
3	Nisam Antara	5,083	7,108	2,727
4	Kuta Makmur	38,020	39,577	35,631
5	Syamtalira Bayu	5,460	7,035	3,008
6	Geureudong Pase	12,125	13,543	9,685
7	Samudera	3,223	0,000	4,502
8	Meurah Mulia	3,801	5,356	1,376
9	Tanah Luas	3,760	5,720	1,341
10	Matang Kuli	0,487	2,942	2,288
11	Pirak Timu	2,462	4,508	0,095
12	Lhoksukon	33,465	34,960	31,061
13	Baktiya	14,666	16,260	12,259
14	Tanah Jambo Aye	16,849	18,421	14,443
15	Cot Girek	38,765	40,297	36,371
16	Langkahan	32,535	34,031	30,131
17	Baktiya Barat	1,408	1,956	2,708
18	Paya Bakong	1,613	3,875	0,883
19	Nibong	2,292	1,008	3,941
20	Simpang Kramat	2,965	5,189	0,693

Table 9 presents the Euclidean distances calculated between each subdistrict and the initial centroids for three clusters in the conventional K-Means model, with Samudera, Langkahan, and Baktiya serving as initial centroids (C1, C2, and C3, respectively). The table reveals how closely each subdistrict aligns with these centroids, where smaller distances indicate a higher likelihood of a subdistrict belonging to that particular cluster. For instance, the subdistrict Samudera has a

distance of 0.000 to C1, affirming it as the initial centroid for that cluster. Similarly, Baktiya shows relatively small distances to its designated centroid, C3, while Langkahan displays minimal distance to C2, anchoring each as central points in their respective clusters. Some subdistricts, such as Pirak Timu and Baktiya Barat, show low distances to multiple centroids, suggesting they may lie near the boundaries of these clusters and may shift in subsequent iterations. This table provides insight into the initial grouping structure. K-Means will iteratively adjust centroids based on these calculated distances to minimize within-cluster variance, ultimately forming clusters with greater homogeneity. The initial distances guide the model's iterative process, influencing cluster composition and convergence in the final clustering result. The clustering results are presented in Table 10. Table 10 shows the clustering results for each subdistrict in North Aceh, based on the conventional K-Means model, with each subdistrict assigned to one of three clusters (Cluster 1, Cluster 2, and Cluster 3). These clustering results can support targeted strategies for developing the oil palm sector in North Aceh, as shown in Figure 5.

Table 10 Conventional K-Means Clustering Results for Subdistricts

Subdistricts	Cluster
Samudera	1
Nibong	1
Langkahan	2
Cot Girek	3
Baktiya	2
Nisam Antara	1
Sawang	2
Nisam	2
Kuta Makmur	2
Syamtalira Bayu	2
Geureudong Pase	2
Meurah Mulia	3
Tanah Luas	1
Matang Kuli	1
Pirak Timu	3
Lhoksukon	3
Tanah Jambo Aye	2
Langkahan	2
Paya Bakong	2
Simpang Kramat	2

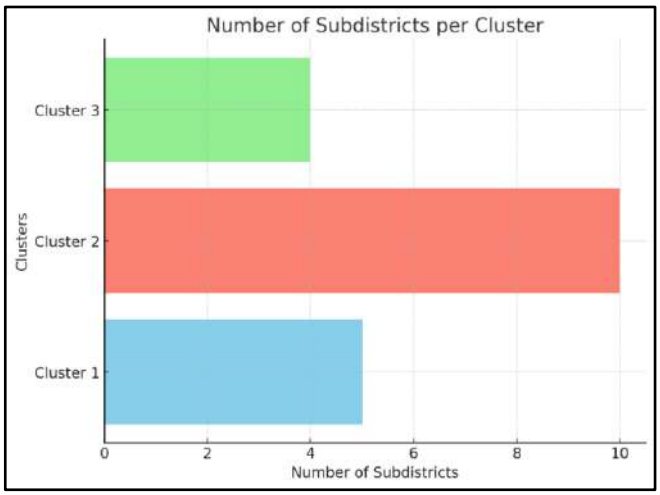


Figure 5 Distribution of Clustering Results with Conventional K-Means

These clusters provide insight into patterns within the region's oil palm sector. Cluster 1 includes subdistricts such as Sawang, Nisam, Geureudong Pase, Baktiya, and Tanah Jambo Aye, suggesting that these areas may share specific characteristics in oil palm productivity or resources that distinguish them from other clusters. Cluster 2, as the largest group, includes subdistricts like Nisam Antara, Samudera, and Meurah Mulia, indicating that this cluster represents the dominant pattern across the data, potentially covering regions with average productivity or typical oil palm-related features. Cluster 3, consisting of Kuta Makmur, Lhoksukon, Cot Girek, and Langkahan, likely represents subdistricts with unique attributes that set them apart from Clusters 1 and 2, possibly due to distinct environmental or infrastructural factors affecting oil palm production. The size of Cluster 2 suggests that it may capture the most prevalent characteristics across North Aceh's oil palm sector.

In contrast, Clusters 1 and 3 may represent more specialized or unique patterns within the industry. For instance, interventions or policies could be tailored to address each cluster's specific needs or strengths, likely more homogeneous within groups than across them. By leveraging these insights, decision-makers can apply targeted approaches to improve productivity, resource allocation, and sustainable practices within the sector, ensuring each cluster receives appropriate support based on its shared characteristics. The visualization of the clustering results using Conventional K-Means is presented in Figure 6.

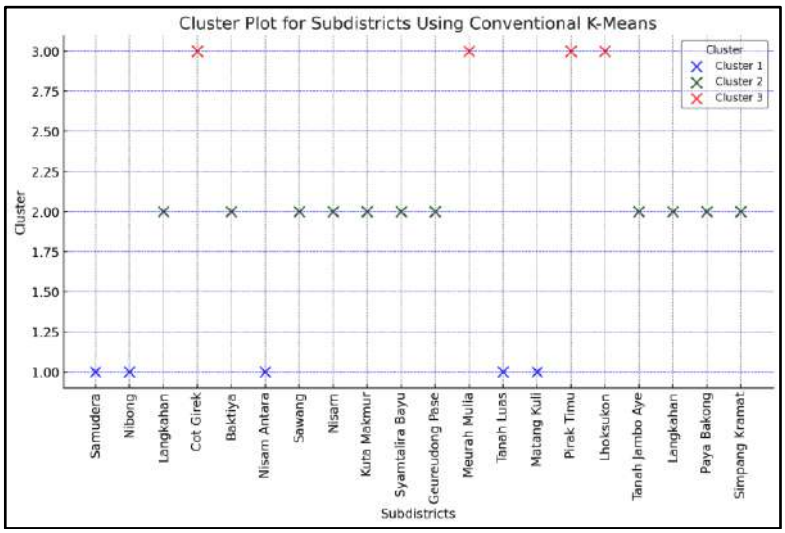


Figure 6 Visualization of the clustering results using Conventional K-Means

3.4 DBI Values and Iterations in Purity K-Means and Conventional K-Means

In evaluating the performance of clustering algorithms, the Davies-Bouldin Index (DBI) is a crucial metric for assessing the quality of the clusters formed. A lower DBI value indicates better cluster separation and cohesion. This section compares the iterations and DBI values of the Purity K-Means and Conventional K-Means algorithms. The analysis highlights the effectiveness of the Purity K-Means approach, demonstrating its superior performance in achieving lower DBI values with fewer iterations, thereby suggesting more efficient clustering, as shown in Table 11.

Table 11 Comparison of Iterations and DBI Values for Purity K-Means and Conventional K-Means

Method	Iterations	DBI Value
Purity K-Means	3	0.30
Conventional K-Means	15	0.45

The data presented in Table 8 highlights a significant improvement in the performance of the Purity K-Means algorithm compared to Conventional K-Means. Specifically, the Purity K-Means achieved its clustering results with only three iterations, while Conventional K-Means required 15 iterations. This reduction in iterations demonstrates the efficiency of the Purity K-Means approach and suggests that it can optimize the K-Means algorithm's performance. Additionally, the Dunn Index (DBI) values further substantiate these findings, with the Purity K-Means recording a lower DBI value of 0.30 compared to 0.45 for the Conventional K-Means. A lower DBI value indicates better clustering quality, confirming that the Purity K-Means method effectively minimizes the clustering overlap while enhancing the distinctiveness of clusters. Overall, these results illustrate that the Purity K-Means algorithm not only streamlines the clustering process but also enhances the overall quality of the results.

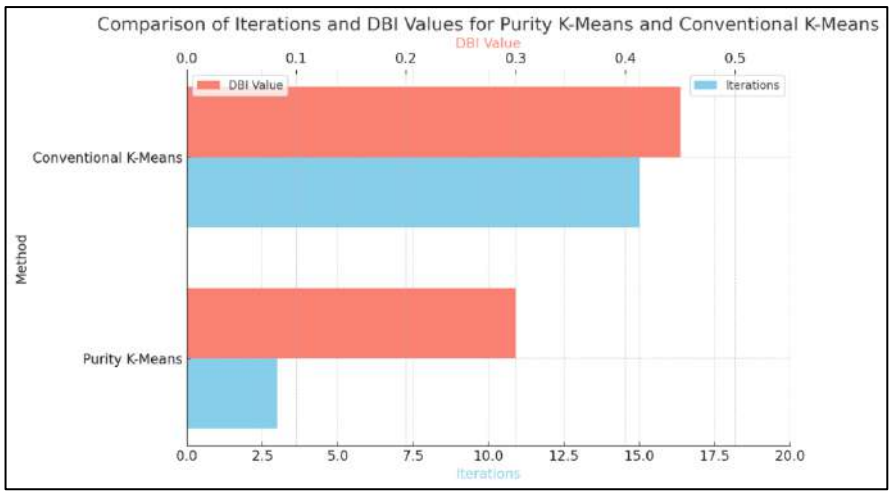


Figure 7 Comparison of Iterations and DBI Values for Purity K-Means and Conventional K-Means

The horizontal bar chart illustrates the superior performance of the Purity K-Means algorithm compared to Conventional K-Means in terms of iterations and Davies-Bouldin Index (DBI) values. The Purity K-Means method achieves a remarkable reduction in iterations, requiring only three compared to 15 for the Conventional K-Means, indicating a more efficient clustering process and faster convergence to optimal solutions. Furthermore, the Purity K-Means demonstrates a lower DBI value of 0.30, in contrast to 0.45 for the Conventional K-Means. This lower DBI signifies better clustering performance, highlighting greater cluster separation and reduced intra-cluster variance. These findings emphasize that the Purity K-Means algorithm optimizes computational efficiency and enhances clustering quality, making it a valuable approach for effective data clustering.

4. CONCLUSIONS

This study successfully enhanced the performance of the K-Means algorithm by integrating the Purity method, focusing on oil palm production regions in North Aceh. The results demonstrated a significant improvement in clustering efficiency, as the Purity K-Means approach reduced the number of iterations required for convergence from 15 in conventional K-Means to just 3. Additionally, the Davies-Bouldin Index (DBI) value indicated a notable enhancement in cluster quality, decreasing from 0.45 in conventional K-Means to 0.30 in the Purity K-Means method.

The clustering analysis identified three distinct clusters within the subdistricts of North Aceh. Cluster 1 included subdistricts such as Sawang, Nisam, Geureudong Pase, Baktiya, and Tanah Jambo Aye, indicating shared characteristics in oil palm productivity. Cluster 2, the largest, comprised subdistricts such as Nisam Antara, Samudera, and Meurah Mulia, representing the region’s predominant production patterns. Finally, Cluster 3 included Kuta Makmur, Lhoksukon,

Cot Girek, and Langkahan, likely reflecting distinctive attributes shaped by specific environmental or infrastructural factors.

These findings provide valuable insights for developing targeted strategies to enhance the oil palm sector in North Aceh and offer potential applications in other regions with similar conditions. By understanding each cluster's unique characteristics and needs, policymakers and stakeholders can implement tailored interventions to optimize productivity, resource allocation, and sustainability. Integrating the Purity method with the K-Means algorithm demonstrates significant potential for improving clustering outcomes in agricultural data analysis and related fields.

REFERENCES

Ariyanto, Y., Sabilla, W. I., & As Sidiq, Z. S. (2024). Recommendation System for Clustering to Allocate Classes for New Students Using the K-Means Method. *Compiler*, *13*(1), 27. https://doi.org/10.28989/compiler.v13i1.1962

Bhatti, M. A., Zeeshan, Z., M.S., S., Bhatti, U. A., Khan, A., Ghadi, Y. Y., Alsenan, S., Li, Y., Asif, M., & Afzal, T. (2024). Advanced Plant Disease Segmentation in Precision Agriculture Using Optimal Dimensionality Reduction with Fuzzy C-Means Clustering and Deep Learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *17*, 18264–18277. https://doi.org/10.1109/JSTARS.2024.3437469

Cebolla-Alemany, J., Macarulla Martí, M., Viana, M., Moreno-Martín, V., San Félix, V., & Bou, D. (2024). Optimizing Indoor Air Models Through K-Means Clustering of Nanoparticle Size Distribution Data. *Building and Environment*, *266*, 112091. https://doi.org/10.1016/j.buildenv.2024.112091

Dinata, R. K., Adek, R. T., Hasdyna, N., & Retno, S. (2023). K-Nearest Neighbor Classifier Optimization Using Purity. *AIP Conference Proceedings*, *2431*(1). https://doi.org/10.1063/5.0117058/2906121

Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., & Akinyelu, A. A. (2022). A Comprehensive Survey of Clustering Algorithms: State-of-the-Art Machine Learning Applications, Taxonomy, Challenges, and Future Research Prospects. *Engineering Applications of Artificial Intelligence*, *110*, 104743. https://doi.org/10.1016/j.engappai.2022.104743

Hasdyna, N., & Dinata, R. K. (2024). Comparative Analysis of K-Medoids and Purity K-Medoids Methods for Identifying Accident-Prone Areas in North Aceh Regency. *Scientific Journal of Informatics*, *11*(2), 263–272. https://doi.org/10.15294/SJI.V11I2.3433

Henderi, H., Fitriana, L., Iskandar, I., Astuti, R., Arifandy, M. I., Hayadi, B. H., Mesran, M., Chin, J., & Kurniawan, A. (2024). Optimization of Davies-Bouldin Index with K-Medoids Algorithm. *Science and Technology Research Symposium 2022*, *3065*(1), 030002. https://doi.org/10.1063/5.0225220/3311944

Kouadio, K. L., Liu, J., Liu, R., Wang, Y., & Liu, W. (2024). K-Means Featurizer: A Booster for Intricate Datasets. *Earth Science Informatics*, *17*(2), 1203–1228. https://doi.org/10.1007/S12145-024-01236-3/METRICS

Li, M., Frank, E., & Pfahringer, B. (2023). Large Scale K-Means Clustering Using GPUs. *Data Mining and Knowledge Discovery*, *37*(1), 67–109. https://doi.org/10.1007/S10618-022-00869-6/TABLES/22

Majumdar, P., Bhattacharya, D., Mitra, S., Solgi, R., Oliva, D., & Bhusan, B. (2023). Demand Prediction of Rice Growth Stage-Wise Irrigation Water Requirement and Fertilizer Using Bayesian Genetic Algorithm and Random Forest for Yield Enhancement. *Paddy and Water Environment*, *21*(2), 275–293. https://doi.org/10.1007/S10333-023-00930-0/METRICS

Naz, H., Saba, T., Alamri, F. S., Almasoud, A. S., & Rehman, A. (2024). An Improved Robust Fuzzy Local Information K-Means Clustering Algorithm for Diabetic Retinopathy Detection. *IEEE Access*, *12*, 78611–78623. https://doi.org/10.1109/ACCESS.2024.3392032

Retno, S., Hasdyna, N., & Yafis, B. (2024). K-NN with Purity Algorithm to Enhance the Classification of the Air Quality Dataset. *Journal of Advanced Computer Knowledge and Algorithms*, *1*(2), 42–46. https://doi.org/10.29103/jacka.v1i2.15890

Rezaee, L., Davatgar, N., Moosavi, A. A., & Sepaskhah, A. R. (2023). Implications of Spatial Variability of Soil Physical Attributes in Delineating Site-Specific Irrigation Management Zones for Rice Crop. *Journal of Soil Science and Plant Nutrition*, *23*(4), 6596–6611. https://doi.org/10.1007/S42729-023-01513-Y/METRICS

Ros, F., Riad, R., & Guillaume, S. (2023). PDBI: A Partitioning Davies-Bouldin Index for Clustering Evaluation. *Neurocomputing*, *528*, 178–199. https://doi.org/10.1016/j.neucom.2023.01.043

Thakur, B., & Kaur, S. (2024). The Role of Artificial Intelligence in Biofertilizer Development. In *Metabolomics, Proteomics and Gene Editing Approaches in Biofertilizer Industry* (pp. 157–176). Springer Nature Singapore. https://doi.org/10.1007/978-981-97-2910-4_9

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 16 – 32
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Predicting Olympic Medal Trends for Southeast Asian Countries Using the Facebook Prophet Model

Bagus Al Qohar [(1)*], Yulizchia Malica Pinkan Tanga [(2)], Putri Utami [(3)], Maylinna Rahayu Ningsih [(4)], Much Aziz Muslim [(5)]
[1,2,3,4] Department of Computer Science, Universitas Negeri Semarang, Semarang, Indonesia
[5] Faculty of Technology Management, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia
e-mail :
{bagusximipa6,yulizchiamalica,utamiputri575,maylinnarahayuningsih}@students.unnes.ac.id,
gp200017@student.uthm.edu.my.
* Corresponding author.
This article was submitted on 18 October 2024, revised on 5 November 2024, accepted on 6 November 2024, and published on 31 January 2025.

Abstract

The Olympics is a world sporting event held every four years and is a meeting place for all athletes worldwide. The Olympics are held alternately in different countries. The Olympics were first held in Athens in 1896 and have now reached the 33rd Olympics, which will be held in Paris in 2024. Much work has been done to develop prediction models emphasizing improving accuracy to predict Olympic outcomes. However, low-performance regression algorithms are the main problems with prediction. By integrating custom seasonality with the Facebook-Prophet prediction model, this study aims to increase the accuracy of Olympic prediction. The proposed new model involves several steps, including preparing the data and initializing and fitting the Facebook-Prophet model with several parameters such as seasonal mode, annual seasonality, and prior scale. The model is tested using the Olympic dataset (1994–2024). The evaluation results show that this prediction model can provide a good value in predicting the total medals earned. On the Olympic Games (1994-2024) dataset, the model has a very low error MAE, MSE, and RMSE and has an R2 score of 0.99, which is close to perfect. This research shows that the model is effective in improving prediction accuracy.

Keywords: Custom Seasonality, Facebook-Prophet, Forecasting, Olympic Medals, Time Series

Abstrak

Olimpiade adalah acara olahraga dunia yang diadakan setiap 4 tahun sekali dan merupakan tempat pertemuan bagi semua atlet di seluruh dunia. Olimpiade diadakan secara bergantian di berbagai negara. Olimpiade pertama kali diadakan di Athena pada tahun 1896 dan sekarang telah mencapai Olimpiade ke-33, yang akan diadakan di Paris pada tahun 2024. Untuk memprediksi hasil Olimpiade, banyak upaya telah dilakukan untuk mengembangkan model prediksi dengan penekanan pada peningkatan akurasi. Namun, algoritma regresi berkinerja rendah adalah masalah utama dalam prediksi. Dengan mengintegrasikan musiman khusus dengan model prediksi Facebook-Prophet, penelitian ini bertujuan untuk meningkatkan akurasi prediksi Olimpiade. Model baru yang diusulkan melibatkan beberapa langkah, termasuk menyiapkan data, inisialisasi, dan menyesuaikan model Facebook-Prophet dengan beberapa parameter seperti mode musiman, musiman tahunan, dan skala sebelumnya. Model ini diuji dengan menggunakan dataset Olimpiade (1994-2024). Hasil evaluasi menunjukkan bahwa model prediksi ini dapat memberikan nilai yang baik dalam memprediksi total medali yang diperoleh. Pada dataset Olimpiade (1994-2024), model ini memiliki error MAE, MSE, dan RMSE yang sangat rendah serta memiliki nilai R2 sebesar 0.99, yang mendekati sempurna. Penelitian ini menunjukkan bahwa model efektif dalam meningkatkan akurasi prediksi.

Kata Kunci: Musiman Khusus, Facebook-Prophet, Peramalan, Medali Olimpiade, Deret Waktu

1. INTRODUCTION

Almost a century ago, the Olympics brought athletes from all over the world to compete for medals. Many nations have made significant Olympic progress, earning respect and recognition for their achievements (Herzog, 2024; James, 2023; Theodorakis et al., 2024). Southeast Asia currently receives up to 104 medals. Not only will strategic planning and resource allocation benefit from an analysis of the various elements that influence the success of the Olympics (Badoni et al., 2023), but future achievements will also be enhanced. This paper clarifies the distribution of medals in individual Olympic sports. The aim is to understand how country variables, including population and economic size, influence the share of the medal in various individual sports.

Another study looks at how public sports expenditure affects the Olympic medal count. (Wu et al., 2023). This study seeks to determine whether public sports investment increases in line with the Olympic success rate of a nation. It also investigates (Wen & Wang, 2020) whether the nations' climate determines the success of the Olympic Games in a significant way. The aim is to determine whether there is any correlation between climate origins and a nation's success or specialization in one type of sport throughout the six editions of the Olympic Games taken under review between 1996 and 2016 (Scelles et al., 2020). In order to improve current approaches by considering economic, demographic, and historical factors, this paper reevaluates the estimate of the number of medals that countries will win at the Summer Olympics. Another study investigates elements that support or hinder a nation's Olympic performance. (Rewilak, 2021). The objective is to identify the primary and less important elements that influence the medal count of a country, as well as to investigate the causes of different degrees of success among different countries. Predictive modeling methods, including Prophet, have recently been used to identify and project various results, from sports results to Bitcoin projections. (Cheng et al., 2024). Predictive modeling provides important information that allows many stakeholders, including countries, to make informed decisions.

In line with this, (Asha et al., 2023) Examined Olympic Games performance using data analytics, spotting public investment and economic power as the main determinants of a nation's medal count. Similarly, (Badoni et al., 2023) Compared machine learning algorithms to forecast Olympic medal counts and identified random forest and gradient boost as quite successful models. With XGBoost turning out to be the most accurate, (Sagala & Amien Ibrahim, 2022) investigated the efficacy of several boosting techniques for estimating Olympic medals. (Xinyi & Chenglong, 2022a) Visual analytics also examines trends in Olympic medal distribution, highlighting the relevance of geographic and population elements in medal success. Recent (Jia et al., 2022a) analysis of public opinion worldwide during four Olympic Games (2008–2022). The research revealed that geopolitical and social elements significantly affect public opinion and sentiment about the Olympics, underscoring how perspective can help shape global expectations for national Olympic performances.

Furthermore, as underlined in the research by (Agyemang et al., 2023), Predictive analytics are needed to improve national Olympic readiness. This study uses predictive models to investigate how nations might maximize their resources and increase their chances of success. Furthermore, research on predictive modeling has demonstrated the success of several strategies, including time-series analysis, to project several results (Satrio et al., 2021; Wulandari et al., 2021). For example, the adaptability of ARIMA and Prophet was shown by forecasting COVID-19 cases in Indonesia, proving their use. Further underscoring the resilience of these forecasting methods, (Angelo et al., 2023) offered a comparison of the ARIMA and Prophet algorithms in estimating Bitcoin prices. (Li et al., 2021) also evaluated two-stage network structures with the 2018 Winter Olympic Games, helping to clarify how various modeling techniques evaluate Olympic performance.

Thus, by guiding nations in the wise use of limited resources, predictive models such as Facebook Prophet help them better prepare for challenges (Agyemang et al., 2023). Predictive analysis is

today a useful tool for evaluating past performance and projecting future outcomes. Although building the predictive model is easy, the complexity of the data makes achieving dependability in results difficult (Santos Arteaga et al., 2024). From statistical approaches to machine learning, the literature (Chowdary et al., 2024; Lei et al., 2024) has investigated many ways to project sports performance. Low-quality data could potentially lead to errors due to the poor performance of the categorization algorithm. Therefore, we must improve the models to increase the accuracy of the prediction.

In the present paper, we investigate the optimization of the predictive model using a historical performance analysis of several Southeast Asian countries and future projections using the Prophet model. The choice of the Prophet model is justified due to its ability to capture patterns in historical data effectively, provide highly accurate predictions with low error values, and explain variations in medal acquisition almost perfectly, making it a valuable tool for policymakers and coaches to enhance future performance in Olympic events. Unlike ARIMA and other traditional time series models, Prophet can adjust for seasonality and external variables that impact athletes' performances, leading to more accurate forecasts. Seasonality in Olympic forecasting refers to the recurring patterns and trends that occur in the context of the Olympic Games. Seasonality significantly influences medal predictions by allowing the model to capture recurring patterns, enhance forecast accuracy, and provide insights into the factors affecting Olympic performance. This understanding is vital for countries planning their strategies for future Olympic events. The research contributes by providing more accurate predictions of Southeast Asian countries. Olympic medal counts and effectively guiding strategic planning and resource allocation for future Olympic games based on accurate projections for countries like Thailand, Indonesia, Malaysia, Singapore, and the Philippines, highlighting improvement areas and ongoing support for sports programs.

2. METHODS

Using the Prophet model, Figure 1 shows the approach to estimate the total number of medals Southeast Asia will score at the Olympics. The data is then compiled to ensure they are in a suitable format for study. (Guo et al., 2021) To increase the accuracy of the forecasts, the Prophet model is then started with custom parameters that consider both additional seasonal components and annual seasonality.

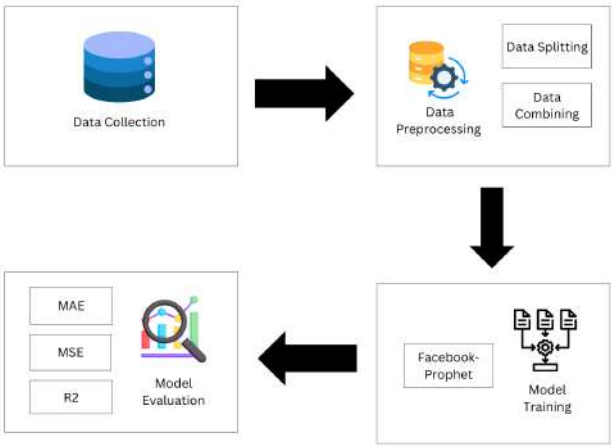


Figure 1 Research Method

2.1 Data Collection

Data collection for this study involved collecting Olympic medal counts from various Olympic Games, both summer and winter, ranging from 1994 to 2024 (Ismail, 2024). Medal information was taken from multiple CSV files, including information from the Olympic Games in Atlanta in

1996, Beijing in 2008, Athens in 2004, Torino in 2006, Paris in 2024, and other locations. The files include the total number of gold, silver, and bronze medals that the National Olympic Committee (NOC) of each nation has won. There have been 879 entries in total, representing the nation's medal total over several years and sports. This dataset is the foundation for examining patterns and trends in Olympic achievements.

For accurate forecasting results, the quality of historical data is crucial. However, the period covered by this data is diverse, which poses various challenges, including the climate factor. This climate factor is an external factor that can affect the predicted medal results because it impacts athlete performance, such as an advantage for local athletes or those accustomed to similar climates, shifts in sleep and training patterns, and others. Table 1 summarizes the data collection for each Olympic event and shows the number of entries in each dataset.

Table 1 Table Data Collection

Year	Olympic event	Number of Entries
1994	Lillehammer (Winter)	22
1996	Atlanta (Summer)	78
1998	Nagano (Winter)	24
2000	Sydney (Summer)	79
2002	Salt Lake City (Winter)	24
2004	Athens (Summer)	74
2006	Torino (Winter)	26
2008	Beijing (Summer)	87
2010	Vancouver (Winter)	26
2012	London (Summer)	86
2014	Sochi (Winter)	26
2016	Rio (Summer)	85
2018	PyeongChang (Winter)	30
2020	Tokyo (Summer)	92
2022	Beijing (Winter)	29
2024	Paris (Summer)	91
Total		**879**

2.2 Data Preprocessing

To manage data from many sources, we have created several significant data preprocessing methods (Tawakuli et al., 2024). This starts with gathering CSV data from a particular directory with Olympic records. We filter these CSV files depending on their.csv extension to handle the pertinent ones. Automated file loading guarantees extracting all pertinent files; hence, it is one of the most important techniques for handling large databases. Moreover, this script is dynamic since it accesses and lists every file in the specified directory using the OS.listdir tool. As noted by (Phan et al., 2021), who emphasized the importance of dynamic and automated file-handling techniques in wind power forecasting, this approach provides greater flexibility in managing variable datasets over time.

Then, the code takes care of the important chore of deleting the year from every one of these file names. The code cleverly reads the filenames for the year and adds them as a new column in every matching data frame, considering that the dataset comprises several files, each corresponding to a different year of Olympic data. This will maintain the data chronologically and simplify time-based analyses, including trend tracking and outcome prediction. Once we include the year column, downstream analyses or modeling jobs will find the dataset more consistent and manageable. Like the year-specific feature engineering used by (Ding et al., 2024), our approach helps guarantee the integrity and accuracy of time-based data studies for carbon emissions forecasting.

Using pandas.concat (Dong et al., 2024), the last phase of this preprocessing process concatenates the individual data frames into one complete data frame. Similarly, the code concatenates data frames, aggregating the information from many sources into a single dataset fit for extensive study. (Yin et al., 2024) it has developed a three-stage data preprocessing plan that aligns with this method, demonstrating the efficacy of combining multiple datasets for long-term freight market prediction. Figure 2 illustrates the merging of multiple datasets into a single combined data frame.

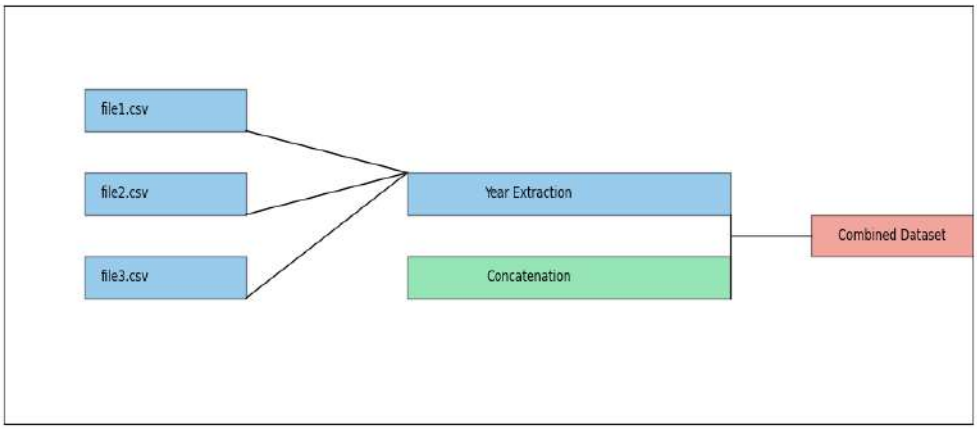


Figure 2 Process of Merging Multiple Datasets Into a Combined Data Frame

2.3 Model Training

We design the key steps in model training using the Prophet algorithm to capture trends and seasonal patterns forecasting Southeast Asia's total Olympic medals. The prepare_prophet_data custom function first structures the Olympic dataset in a format compatible with Prophet, using the 'total' medal column, which includes gold, silver, and bronze medals. Prophet is particularly suited for handling time-series data with irregular spacing and missing values, making it ideal for long-term forecasting tasks, as demonstrated in various forecasting domains (Annapoorna et al., 2024; Gautam et al., 2023; Mousa et al., 2023).

After preparing the data, we initialise the Prophet model with certain parameters to improve the forecast quality. For example, yearly_seasonality = True allows the model to capture recurring annual patterns, such as those seen in the Olympics. We also set changepoint_prior_scale = 0.8 so that the model can respond more flexibly to trends, such as significant performance spikes in countries like Indonesia. In addition, we add custom quadrennial seasonality to reflect the Olympic cycle that occurs every four years so that the model can account for the unique periodicity of this event. This method uses the custom_seasonality parameter, which helps the model to recognise and capture specific quadrennial patterns more effectively.

The four-year seasonality in Olympic forecasting refers to the recurring patterns and trends that occur every four years in the context of the Olympic Games. This cycle is crucial for understanding and predicting medal achievements over time, considering the quadrennial nature of the event. (Gong et al., 2020; Verghese et al., 2021). In the Prophet model, the custom four-year seasonality is implemented by incorporating a specific parameter that accounts for the unique periodicity of the Olympic cycle. This feature allows the model to adjust its forecasts based on the cyclic nature of the Olympics, capturing the long-term trends and patterns in the medal data. By considering the four-year seasonality, the forecasting model can better predict medal counts by accounting for the historical patterns that repeat every Olympic cycle. This approach enhances the accuracy and reliability of the forecasts, providing valuable insights for strategic planning and resource allocation in the context of the Olympics. In summary, seasonality significantly influences medal predictions by allowing the model to capture recurring patterns, enhance forecast accuracy, and

provide insights into the factors affecting Olympic performance. This understanding is vital for countries planning their strategies for future Olympic events.

This training aims to forecast the total medal count for the next four Olympic cycles. The code also includes a visualization step that plots the actual and predicted total medals, allowing for a visual comparison of the forecasted outcomes. The plotting process results in Figure 3 demonstrate the successful adaptation of Prophet's forecasting techniques for this task. This plot is useful for evaluating the model performance in forecasting. Plotting predictions against actual results allows one to quickly assess how well the model captures historical trends and whether the future forecasted trend follows reasonable patterns. Moreover, including visual grids and labels enhances the plot's clarity and informational value, making it suitable for presentation.

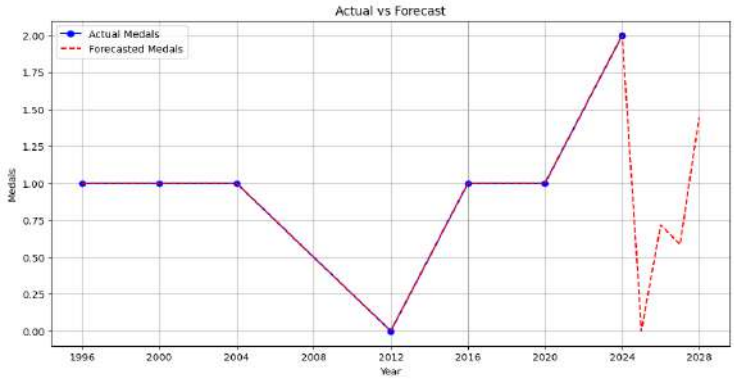


Figure 3 Plot of The Model Training Result

2.4 Model Evaluation

Confirming that the forecasts are accurate and reliable depends on assessing the model's performance first. The measures offer an insightful analysis of the performance of the Prophet model in terms of predicting the total Olympic medal count for Southeast Asian countries. Ignoring its direction, the mean absolute error (MAE) only considers the average size of the errors, allowing for a clear awareness of the difference between the predictions and the actual value (Mahajan & Shrivastav, 2023; Rajesh & Saravanan, 2022). Due to the square difference between predicted and actual values, MSE and RMSE highlight larger errors (Karunasingha, 2022; Qi et al., 2020). Therefore, these measures are especially sensitive to major forecast deviations.

In the fields of predictive modeling and data analysis, MAE is a metric that is frequently utilized. This equation provides a straightforward method for determining the accuracy of predictions by calculating the average magnitude of errors that occur in a set of predictions by using the equation. MAE is represented by the Equation (1).

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \quad (1)$$

The total of these absolute errors across all of the data points is computed by this equation, which calculates the total. Obtaining the mean average absolute error can be accomplished by dividing this total by n, which is the number of measurements. The value that is predicted to be the number of observations in the dataset is denoted by the symbol yi. The corresponding true or observed value for the same observation is denoted by the symbolic value xi. The total number of data points contained in the dataset is denoted by the letter n. Absolute difference, denoted by the notation |yi-xi|, is a measurement that determines the degree to which each prediction deviates from its actual value. This ensures that all errors are regarded as positive values.

The mean squared error (MSE) is a metric that is frequently utilized for the purpose of evaluating the precision of predictions made by regression and forecasting models. Therefore, it is sensitive to larger errors because it quantifies the average squared difference between the values that were observed and the values that were predicted. The equation for MAE is shown in Equation (2)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (2)$$

In order to provide a measurement of the overall error in the predictions, this equation can be used to aggregate the squared deviations across all of the data points. The mean or average squared error can be calculated by dividing this total by the number of observations. Yi is the value that has been observed or is the actual value for the number of data points that are contained in the dataset. For the same data point, the value that is predicted is denoted by the symbol $\hat{Y}_i$. n represents the total number of data points that are contained within the dataset. The squared difference, denoted as $(Y_i - \hat{Y}_i)^2$, is a statistical measure that determines the squared deviation for every prediction. This technique amplifies the impact of larger errors.

R^2 is a statistical metric used to evaluate the goodness of fit for regression models. It quantifies the proportion of variation in the dependent variable that is explained by the independent variables in the model. Values closer to 1 indicate that the model is more fitted. The equation for R^2 showed in Equation (3)

$$R^2 = 1 - \frac{RSS}{TSS} \quad (3)$$

The Residual Sum of Squares, also known as RSS, is a statistical technique that allows for the quantification of the model's unexplained variance by measuring the total squared differences between the predicted and observed values. The RSS equation is shown in Equation (4).

$$RSS = \sum_{i=1}^{n}(yi - f(xi))^2 \quad (4)$$

Using this equation, the squared residuals are accumulated across all of the data points that are included in the dataset. The RSS that is produced is a reflection of the overall magnitude of the prediction errors. RSS values that are smaller indicate that the model is a better fit for the data, whereas RSS values that are larger suggest that there are greater deviations between the values that observed and those that predicted. The value of yi is the number of values that have been observed in the dataset that the model is attempting to predict. The predicted value for yi is denoted by the symbol f(xi), which is produced by the regression model through the utilization of the input. xi. n is the total number of data points, which serves as the limit of the summation from the highest possible value. The squared term, denoted as $(yi - f(xi))^2$, is used to compute the square of the residual, which is the difference between the values that were observed and those that were predicted, for every single data point.

TSS is a statistical measure that is utilized for the purpose of quantifying the total variation that exists within a dataset. As a baseline for determining how well a regression model fits the data, it is a representation of the overall dispersion of observed values around their point of origin. The equation for RSS is shown in the Equation (5).

$$TSS = \sum_{i=1}^{n}(yi - \bar{y})^2 \quad (5)$$

Using this equation, the total squared standard deviation (TSS) is calculated by averaging the squared deviations of all the observations. It is possible for a regression model to provide an

explanation for the total variability in the dataset, which is reflected by this metric. yi represents the number of values that have been observed in the dataset. It is commonly known as the sample mean, and the symbol ̲ represents the average of all the values that have been observed in the dataset. In this dataset, the total number of observations is denoted by the letter n. In order to determine the distance between each point and the average, the squared term $(yi - \underline{\bar{y}})^2$ is utilized to calculate the squared deviation of each observed value from the mean.

This method extends the mean squared error (MSE) by further converting the units back to the original scale of the data, thus improving the interpretability of the results. The lower the RMSE value, the better the forecast value. In contrast, the R-squared returns indicate how the model predictions align with the actual data model predictions. Values closer to 1 indicate that the model is more fitted. When comparing different models or evaluating a model's ability to explain data variation, this metric proves particularly useful. These metrics combined provide complete insight into overall accuracy and areas where the model may need improvement.

Once the evaluation is complete, the results provide a clear understanding of the model's performance with Indonesia's total Olympic medals. Furthermore, the process enlightens advanced fine-tuning or adjustments in the parameter optimization or the data preprocessing step. High readings of the RMSE or MSE indicate that the model likely struggles to predict several medal trends, which may require revisiting the settings concerning seasonality or incorporating more relevant features into the data.

3. RESULTS AND DISCUSSION

This research focuses on forecasting trends in Olympic medal achievements for Southeast Asian countries comprising Indonesia (INA), Thailand (THA), Malaysia (MAS), Singapore (SIN), and the Philippines (PHI). We use historical data from medal records to identify patterns and make the necessary forecasts, which could help these nations plan appropriately for future Olympic events. The insights derived from these predictions will help highlight potential areas for improvement and investment in your sports programs. In analysing these predictions, we use a robust prediction model known as the Facebook Prophet. It is quite flexible for handling most types of time series data. We configured the model to accoannualannually, weekly, a seasonalitiesnd daily, to highlight the inherent patterns within the series. More importantly, a unique seasonal component was introduced to capture those specific variations within the Olympic cycle. This allows us to produce more customized and meaningful forecasts that align with the trend of medal achievements at the Olympics.

3.1 Indonesia (INA) Olympic Medal Trends

The model emphasizes the total number of medals that Indonesia has acquired during several Olympic eras based on the results of the medal prediction for the country. Prophet's capacity to capture seasonal patterns and long-term trends, which thus improves Indonesia's future medal projections, was the main factor in selecting him. By contrasting the actual medal data acquired by Indonesia with the total expected medal results from the model, Figure 4 offers a better image of the outcome. The solid blue line of the graph shows Indonesia's real medal count over every Olympic cycle. In contrast, the red dotted line shows the expected model results. This visualization helps to see how the forecasts match reality and to spot areas where reality and the predictions could differ.

Visualization indicates that Indonesia has developed a trend pattern in its medal tally, with significant changes occurring in certain years. For example, the effectiveness of the training program or approach during those years, compared to previous periods, could explain the significant increase in medals between 2000 and 2020. For this trend, one could give a detailed view of showing research of different variables that may have caused differences in these medal changes, for example, how policy and sporting rules are changing, the preparation of athletes, and investment in sport.

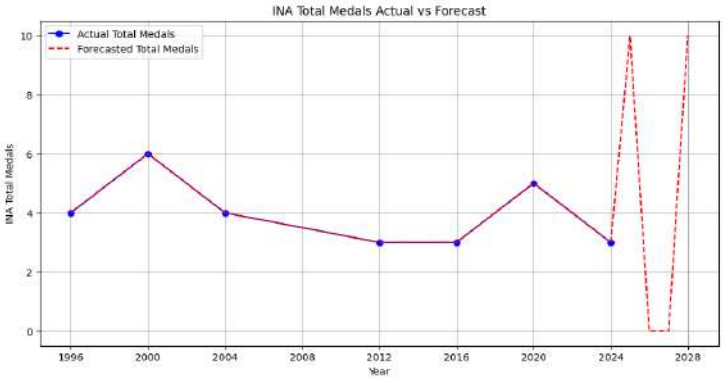


Figure 4 Indonesia Total Medals Actual vs. Predicted

It also involves identifying years of depreciation, such as Indonesia's performance at the 2012 and 2016 Olympics. For example, the reasons could range from a lack of government support to ineffective strategy changes. Understanding the ups and downs of these trends is crucial for future strategy design. The model in Table 2 was evaluated by comparing its performance with other models using four evaluation metrics: mean absolute error, mean squared error, root mean squared error, and R^2.

Table 2 Metric Evaluation of Indonesian Total Medals: Actual vs. Predicted

Metric	Result
MAE	0.0001
MSE	0.0
RMSE	0.0001
R2	0.9999999951

The same period is represented by the evaluation results, where the Prophet model gives very accurate predictions with very low values for MAE, MSE, and RMSE then high values close to 1 for R^2. This indicates that the model almost perfectly explains Indonesia's variation in medal acquisition. This assessment concludes that the Prophet model has successfully accurately predicted Indonesia's medal wins. The accuracy of this model confirms that the Prophet model's use of season and trend components effectively captures patterns in historical data. It can also provide a useful tool for policymakers and coaches to suggest strategies that can help improve Indonesia's future performance in Olympic events.

3.2 Thailand (THA) Olympic Medal Trends

The results of Thailand's medal prediction show notable changes in the gold, silver, bronze, and total medals won over many editions of the Olympic Games. These predictive data provide important new perspectives on Thailand's performance in international events and help identify trends in changes over time. The visualization compares the projected and actual data for Thailand's medal counts. Figure 5 shows the degree of mimicability of the medal trend of the Prophet model. The model's prediction is shown on the dashed line of the graph; the actual medal count is shown on the solid line. This visualization shows that, although the model reasonably captures some variances, generally, the pattern of the actual data and its projections match.
According to trend analysis, Thailand's medal count peaked in 2004 with eight medals: three gold, one silver, and four bronze. Developing athlete training schedules or more government support in that specific year could have helped explain this notable increase. However, the overall medal count dropped in 2020; it rebounded to six in 2024. This trend shows how adaptable and strong Thailand's sport policy is under difficult conditions. We evaluated the performance of the prediction model using R2, mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE), as shown Table 3.

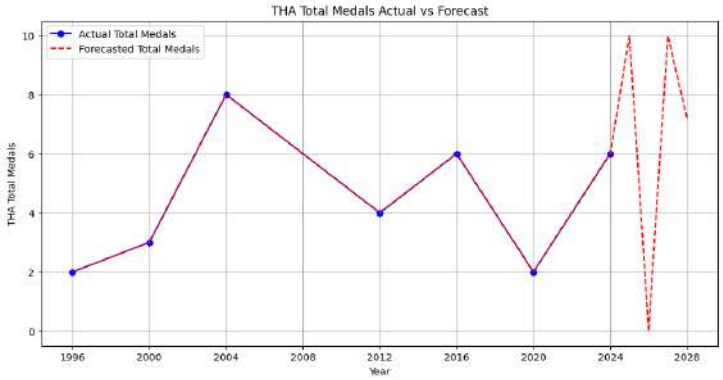


Figure 5 Thailand Total Medals Actual vs. Predicted

Table 3 Table Metric Evaluation for Thailand Total Medals Actual vs. Predicted

Metric	Result
MAE	0.0
MSE	0.0
RMSE	0.0001
R2	0.9999999993

The results of the evaluation metrics show accurate predictions with very low values in estimating the total medals. With MAE and MSE values of 0.0 each, the average prediction error is almost non-existent so the difference between the actual and predicted medal counts is very small. In addition, the very low RMSE of 0.0001 indicates that the prediction error is also very small, and the R^2 value of 0.9999999993 is close to 1, indicating that the model can explain almost perfectly explain the variation in Thailand medal tally. This indicates that the Prophet model successfully accurately predicted Thailand medal tally.

3.3 Malaysia (MAS) Olympic Medal Trends

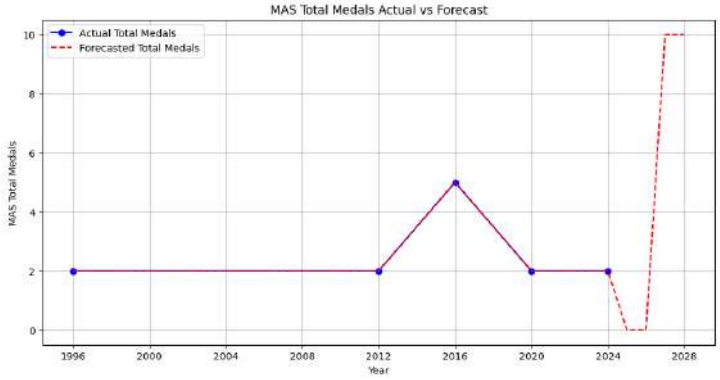


Figure 6 Malaysia Total Medals Actual vs. Predicted

Malaysian medal prediction results (MAS) show notable patterns in the number of silver, bronze, and overall medals acquired by Malaysian athletes over several editions of the Olympic Games. These predictive data help identify changes in Malaysia's performance over time and provide an insightful analysis of her success in international sports events. It also enables one to spot changes in Malaysia's performance over time. Regarding Malaysia's medal counts, the visualization compares the actual data and the anticipated values. Figure 6, for example, shows the precision of replicating the medal trend of the Prophet model. Whereas the solid line shows the actual medal count, the dashed line on the graph shows the model forecast. Using this visualization helps us to see that although the model does a good job of capturing some

fluctuations, the general patterns of the actual data and its forecasts are rather similar to each other.

Over the years, Malaysia's overall medal count has fluctuated; it peaked in 2016 with five medals, four silver and one bronze. The increase could be attributed to improved athlete training schedules or more government support. The medals dropped to two annually in 2020 and 2024. This trend suggests that, despite difficulties, Malaysia's sports performance has shown some consistency. Using four criteria, mean absolute error, mean squared error, root mean squared error, and R^2, the model shown in Table 4 was evaluated against others.

Table 4 Table Metric Evaluation for Malaysia Total Medals Actual vs. Predicted

Metric	Result
MAE	0.0001
MSE	0.0
RMSE	0.0001
R2	0.9999999974

The results of the evaluation metrics show accurate predictions with very low values in estimating the total medals. With MAE and MSE values of 0.0 each, the average prediction error is almost non-existent so the difference between the actual and predicted medal counts is very small. In addition, the very low RMSE of 0.0001 indicates that the prediction error is also very small, and the R^2 value of 0.9999999974, close to 1, indicates that the model almost perfectly explains the variation in Indonesia's medal tally. This shows that the Prophet model successfully accurately predicted Malaysia medal tally. The model's accuracy shows that the seasonal and trend components of the Prophet model can successfully capture the patterns in the historical data. Moreover, this model can be a very useful tool for coaches and policymakers to make plans that can help improve Malaysia's performance in future Olympics.

3.4 Singapore (SIN) Olympic Medal Trends

Singapore's (SIN) medal distribution at different Olympic Games tells an intriguing story about performance swings. Singapore had little success in 2012, winning two bronze medals. The nation won its first gold medal in 2016, a notable achievement on the international sports scene. However, Singapore's medal total dropped to just one bronze in 2024. These findings suggest that Singapore's overall Olympic performance was not entirely consistent. Figure 7 compares the total actual and predicted medal counts for Singapore at multiple Olympic Games, demonstrating the precision of the Prophet model in tracking Singapore's performance trends.

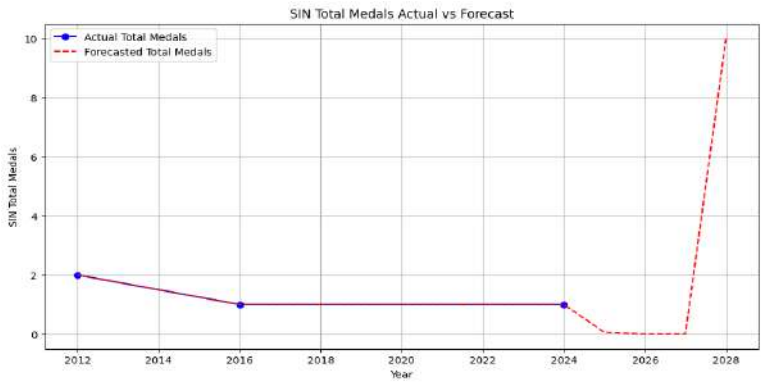


Figure 7 Singapore Total Medals Actual vs. Predicted

In 2016, Singapore achieved a significant turning point by winning its first and only gold medal. We could attribute this achievement to better athlete preparation, improved sports initiatives, or targeted government funding to promote sports in the nation. The lack of medals in the other

categories (bronze or silver) during this period may indicate a focused strategy that produced a win in a single event but had no wider effects. This gold medal win continues to be a high point in Singapore's Olympic history, even with the decline in the following years.

By 2024, Singapore had only won one bronze medal, indicating its performance had deteriorated again. This decrease in the number of medals could indicate difficulties maintaining momentum for 2016. The relatively low number of medals won by Singapore in these three Olympic cycles may suggest that the country's programs for developing athletes, sports infrastructure, or international competitiveness are still in their infancy. Although there is no doubt that success is possible in certain situations, consistency needs work. Table 3 presents the evaluation metrics, including MAE, MSE, RMSE, and R2, which highlight the high precision of the Prophet model in predicting Singapore's total Olympic medal counts.

Table 5 Table Metric Evaluation for Singapore Total Medals Actual vs. Predicted

Metric	Result
MAE	0.0
MSE	0.0
RMSE	0.0
R2	0.9999999973

The evaluation results show that the Prophet model accurately predicts the number of medals won by Singapore with very low MAE, MSE, and RMSE values and R^2 values almost equal to 1. The MAE, MSE and RMSE values of 0.0 each indicate that the average prediction error is almost non-existent, so the difference between the actual and predicted number of medals is very small. The R^2 value of 0.9999999973, close to 1, indicates that the model almost perfectly explains the variation in Singapore medal tally. This shows that the Prophet model successfully accurately predicted Singapore's medal tally. The model's accuracy is demonstrated by the seasonal and trend components of the Prophet model successfully capturing the patterns in the historical data. Moreover, this model can be a useful tool for coaches and policymakers to make plans that can help improve Singapore's performance in future Olympics.

3.5 Philippines (PHI) Olympic Medal Trends

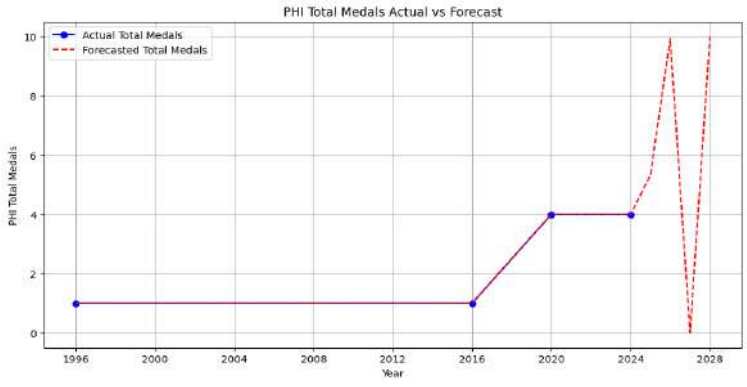


Figure 8 Total Philippines Medals Actual vs. Predicted

The Philippines has shown a clear increasing trend in Olympic medal performance over several years; more recent Games have shown notable gains. From 1996 to 2016, the nation only managed to win one silver medal in each of these years, reflecting its rather low degree of involvement in the international athletic scene. However, these outcomes prepared the ground for a notable shift in the Olympic cycles that followed. Figure 8 shows a comparison between the actual total medal counts for the Philippines and the predicted total medal counts for the nation, highlighting the precision of the Prophet model in catching the upward trend in Olympic

performances. This comparison shows how faithfully the Prophet model captures the growing trend in the Philippines' Olympic performances.

With four overall medals, two silver and one bronze, and a first gold medal, the Philippines reached a noteworthy mark in 2020. Improvement in training programs, infrastructure, and more support for sports development helps explain the increase in performance, highlighting the improved capacities of national athletes. Multiple medals in several categories show the country's competitiveness in different sports disciplines has expanded.

In 2024, the Philippines achieved continued success, securing two gold and two bronze medals, thus maintaining a total medal count of four. The rise in gold medals reinforces the country's growing dominance in specific events, while the stable medal count reflects ongoing momentum. The observed improvement may indicate the effectiveness of long-term strategies designed to develop elite athletes and a sustained emphasis on elevating the nation's performance in international sports events. Table 6 shows the evaluation metrics, namely MAE, MSE, RMSE, and R^2, which indicate the high precision of the Prophet model in forecasting the Olympic medal totals for the Philippines.

Table 6 Table Metric Evaluation for Philippines Total Medals Actual vs. Predicted

Metric	Result
MAE	0.0
MSE	0.0
RMSE	0.0
R2	0.9999999998

The results show that the Prophet model can predict the number of medals the Philippines will receive with very low MAE, MSE, and RMSE values and R^2 values almost equal to 1. The MAE, MSE and RMSE values are each 0.0, indicating that the average prediction error is almost non-existent, so the difference between the actual and predicted number of medals is very small. The R^2 value of 0.9999999998, close to 1, indicates that the model almost perfectly explains the variation in the Philippines medal tally. This shows that the Prophet model accurately predicted the Philippines medal tally. The model's accuracy is demonstrated by the seasonal and trend components of the Prophet model successfully capturing the patterns in the historical data. In addition, this model can be a useful tool for coaches and policymakers to make plans that can help improve the Philippines performance in future Olympics.

3.6 Discussion

Table 2 - 6 shows the total medal prediction performance of Southeast Asian countries based on the evaluation metrics including MAE, MSE, RMSE, and R^2. Overall, the Prophet model shows almost perfect prediction results for each country's analyses: Indonesia, Malaysia, Singapore, Thailand and the Philippines, with very low MAE, MSE and RMSE values reaching 0.0 in some countries. The R^2 values for all five countries were also close to 1, indicating that the model could explain almost all of the variation in medal tally. This study shows that the Prophet model can capture patterns in the medal data of Southeast Asian countries. The model is so accurate that it can help coaches and policymakers improve athletes' performance in future Olympics.

Examining the performance of the Prophet model in terms of Olympic medal count prediction requires careful comparison with previous sports analytics studies. Many earlier studies have used conventional statistical approaches or machine learning algorithms to project medal results, including linear regression and decision trees. However, these methods sometimes struggled to adequately depict the intricate trends and anomalies in the medal data over time. The Prophet model improves accuracy by properly adjusting seasonality and external variables affecting athletes' performances. Based on its evaluation measures, this study shows how well the model can offer more consistent forecasts. Table 7 shows the performance measures of the Prophet

model compared to those of previous studies, highlighting the higher precision achieved in this work.

Table 7 Comparison of The Table With Previous Research

Authors	Data Sample	Summer/Winter	Method	Result
Scelles et al. (2020)	1992 - 2016	Summer	Tobit and Hurdle Econometric Models	The hurdle model provides better predictions for the 2016 and 2020 Olympics, particularly highlighting the significant impact of socio-economic factors and regional variables on medal outcomes.
Rewilak (2021)	1996 - 2016	Summer	Key and less influential factors using the Tobit and hurdle	Population size and host effect are significant determinants of Olympic success.
Jia et al. (2022)	2008 - 2022	Summer	International Public Opinion Analysis Using LDA, TF-IDF, Nave Bayes	The opinions of sports events were more positive in Chinese than in English.
Asha et al. (2023)	2000 - 2020	Summer	Data Analysis for Olympic Performance	The analysis revealed that the United States produced the highest number of athletes for the Olympics, followed by Germany. In contrast, Canada had the lowest number of athletes represented.
Badoni et al. (2023)	2000 - 2020	Summer	Comparative analysis of machine learning algorithms like linear regression and decision trees.	The research compares machine learning algorithms, such as linear regression, decision trees, and support vector machines (SVM), to determine which model provides the most accurate predictions for Olympic medal counts. Decision Tree handles categorical and numerical data efficiently
Proposed Method	1994 - 2024	Summer / Winter	Prediction with the Facebook-Prophet Model	The research forecasts Olympic medal trends for Southeast Asian countries. The Facebook Prophet model effectively predicts medal achievements.

4. CONCLUSIONS

The study of Olympic medal counts in Indonesia, Thailand, Malaysia, Singapore, and the Philippines generally exposes distinctive trends in each nation's performance in recent years. The Prophet model captures these trends, as seen by the close alignment between expected and actual medal totals. The evaluation criteria of the Prophet model in all countries show its effectiveness in catching historical trends and guiding future strategies. Future research on predicting Olympic medal trends could look into various ways to improve the understanding and accuracy of forecasts. One possible approach is to broaden the dataset to incorporate additional

Southeast Asian countries, enabling a more comprehensive analysis of regional trends and patterns in Olympic performance. Also, researchers might look into how socio-economic factors, like government funding and athlete support systems, affect medal outcomes. This could give a better understanding of what influences success at the Olympics.

One possible direction for future research might be to look into how machine learning techniques can be combined with the Prophet model. This could help compare predictive accuracy and determine which methods work best for forecasting. Furthermore, studies examining the long-term impacts of training programs and policy changes on medal counts could provide important information for countries looking to improve their Olympic strategies. By focusing on these areas, future research can help us better understand the dynamics of Olympic sports and improve the predictive models used in this area.

REFERENCES

Agyemang, E. F., Mensah, J. A., Ocran, E., Opoku, E., & Nortey, E. N. N. (2023). Time Series Based Road Traffic Accidents Forecasting via SARIMA and Facebook Prophet Model with Potential Changepoints. *Heliyon*, *9*(12), e22544. https://doi.org/10.1016/j.heliyon.2023.e22544

Angelo, M. D., Fadhiilrahman, I., & Purnama, Y. (2023). Comparative Analysis of ARIMA and Prophet Algorithms in Bitcoin Price Forecasting. *Procedia Computer Science*, *227*, 490–499. https://doi.org/10.1016/j.procs.2023.10.550

Annapoorna, E., Sujil, S. V, S, S., Abhishek, S., & T, A. (2024). Revolutionizing Stock Price Prediction with Automated Facebook Prophet Analysis. *2024 International Conference on Inventive Computation Technologies (ICICT)*, 1307–1314. https://doi.org/10.1109/ICICT60155.2024.10544766

Asha, V., Sreeja, S. P., Saju, B., C S, N., N, P. G., & Prasad, A. (2023). Performance Analysis of Olympic Games Using Data Analytics. *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, 1436–1443. https://doi.org/10.1109/ICEARS56392.2023.10084943

Badoni, P., Choudhary, P., Rudesh, C. P., & Singh, N. T. (2023). Predicting Medal Counts in Olympics Using Machine Learning Algorithms: A Comparative Analysis. *2023 International Conference on Advanced Computing & Communication Technologies (ICACCTech)*, 116–121. https://doi.org/10.1109/ICACCTech61146.2023.00027

Cheng, J., Tiwari, S., Khaled, D., Mahendru, M., & Shahzad, U. (2024). Forecasting Bitcoin Prices Using Artificial Intelligence: Combination of ML, SARIMA, and Facebook Prophet Models. *Technological Forecasting and Social Change*, *198*, 122938. https://doi.org/10.1016/j.techfore.2023.122938

Chowdary, P. H., Kaur, V., Nandeesh, T., Krishan, K., & Kaur, A. (2024). From Athens to Rio: A Comprehensive Data Analysis and Visualization of 120 Years of Olympic History. *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 1–6. https://doi.org/10.1109/ICRITO61523.2024.10522373

Ding, S., Ye, J., & Cai, Z. (2024). Multi-Step Carbon Emissions Forecasting Using an Interpretable Framework of New Data Preprocessing Techniques and Improved Grey Multivariable Convolution Model. *Technological Forecasting and Social Change*, *208*, 123720. https://doi.org/10.1016/j.techfore.2024.123720

Dong, X., Guo, W., Zhou, C., Luo, Y., Tian, Z., Zhang, L., Wu, X., & Liu, B. (2024). Hybrid Model for Robust and Accurate Forecasting Building Electricity Demand Combining Physical and Data-Driven Methods. *Energy*, *311*, 133309. https://doi.org/10.1016/j.energy.2024.133309

Gautam, V., Yadav, V., & Kumar, S. (2023). Diagnosis and Forecast of Murder Rates in India Using Random Forest and Prophet Algorithm. *2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, 173–177. https://doi.org/10.1109/CICTN57981.2023.10141293

Gong, F., Han, N., Li, D., & Tian, S. (2020). Trend Analysis of Building Power Consumption Based on Prophet Algorithm. *2020 Asia Energy and Electrical Engineering Symposium (AEEES)*, 1002–1006. https://doi.org/10.1109/AEEES48850.2020.9121548

Guo, L., Fang, W., Zhao, Q., & Wang, X. (2021). The Hybrid PROPHET-SVR Approach for Forecasting Product Time Series Demand with Seasonality. *Computers & Industrial Engineering*, *161*, 107598. https://doi.org/10.1016/j.cie.2021.107598

Herzog, W. (2024). The Paris 2024 Olympic and Paralympic Games. *Journal of Sport and Health Science*, *13*(6), 717–718. https://doi.org/10.1016/j.jshs.2024.06.003

Ismail, Y. (2024). *Olympic Games (1994-2024)*. https://www.kaggle.com/datasets/youssefismail20/olympic-games-1994-2024

James, M. (2023). Human Rights and the Olympic Charter. *The International Sports Law Journal*, *23*(3), 267–270. https://doi.org/10.1007/s40318-023-00254-5

Jia, K., Zhu, Y., Zhang, Y., Liu, F., & Qi, J. (2022). International Public Opinion Analysis of Four Olympic Games: From 2008 to 2022. *Journal of Safety Science and Resilience*, *3*(3), 252–262. https://doi.org/10.1016/j.jnlssr.2022.03.002

Karunasingha, D. S. K. (2022). Root Mean Square Error or Mean Absolute Error? Use Their Ratio as Well. *Information Sciences*, *585*, 609–629. https://doi.org/10.1016/j.ins.2021.11.036

Lei, Y., Lin, A., & Cao, J. (2024). Rhythms of Victory: Predicting Professional Tennis Matches Using Machine Learning. *IEEE Access*, *12*, 113608–113617. https://doi.org/10.1109/ACCESS.2024.3444031

Li, Y., Liu, J., Ang, S., & Yang, F. (2021). Performance Evaluation of Two-Stage Network Structures with Fixed-Sum Outputs: An Application to the 2018 Winter Olympic Games. *Omega*, *102*, 102342. https://doi.org/10.1016/j.omega.2020.102342

Mahajan, A. S., & Shrivastav, A. (2023). Short Term Load Forecasting Based on Regression Models. *2023 International Conference for Advancement in Technology (ICONAT)*, 1–8. https://doi.org/10.1109/ICONAT57137.2023.10080359

Mousa, M. A., AlMansoori, A. N., & AlAjami, F. A. (2023). A Hybrid PV Power Forecasting Model Implementing Emerging Machine Learning Algorithms: Prophet and Neural Prophet. *2023 Middle East and North Africa Solar Conference (MENA-SC)*, 1–8. https://doi.org/10.1109/MENA-SC54044.2023.10374512

Phan, Q.-T., Wu, Y.-K., & Phan, Q.-D. (2021). An Overview of Data Preprocessing for Short-Term Wind Power Forecasting. *2021 7th International Conference on Applied System Innovation (ICASI)*, 121–125. https://doi.org/10.1109/ICASI52993.2021.9568453

Qi, J., Du, J., Siniscalchi, S. M., Ma, X., & Lee, C.-H. (2020). On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression. *IEEE Signal Processing Letters*, *27*, 1485–1489. https://doi.org/10.1109/LSP.2020.3016837

Rajesh, K., & Saravanan, M. S. (2022). Prediction of Customer Spending Score for the Shopping Mall Using Gaussian Mixture Model Comparing with Linear Spline Regression Algorithm to Reduce Root Mean Square Error. *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 335–341. https://doi.org/10.1109/ICICCS53718.2022.9788162

Rewilak, J. (2021). The (Non) Determinants of Olympic Success. *Journal of Sports Economics*, *22*(5), 546–570. https://doi.org/10.1177/1527002521992833

Sagala, N. T. M., & Amien Ibrahim, M. (2022). A Comparative Study of Different Boosting Algorithms for Predicting Olympic Medal. *2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED)*, 1–4. https://doi.org/10.1109/ICCED56140.2022.10010351

Santos Arteaga, F. J., Di Caprio, D., Tavana, M., Cucchiari, D., Campistol, J. M., Oppenheimer, F., Diekmann, F., & Revuelta, I. (2024). On the Capacity of Artificial Intelligence Techniques and Statistical Methods to Deal with Low-Quality Data in Medical Supply Chain Environments. *Engineering Applications of Artificial Intelligence*, *133*, 108610. https://doi.org/10.1016/j.engappai.2024.108610

Satrio, C. B. A., Darmawan, W., Nadia, B. U., & Hanafiah, N. (2021). Time Series Analysis and Forecasting of Coronavirus Disease in Indonesia Using ARIMA Model and PROPHET. *Procedia Computer Science*, *179*, 524–532. https://doi.org/10.1016/j.procs.2021.01.036

Scelles, N., Andreff, W., Bonnal, L., Andreff, M., & Favard, P. (2020). Forecasting National Medal Totals at the Summer Olympic Games Reconsidered. *Social Science Quarterly*, *101*(2), 697–711. https://doi.org/10.1111/ssqu.12782

Tawakuli, A., Havers, B., Gulisano, V., Kaiser, D., & Engel, T. (2024). Survey: Time-Series Data Preprocessing: A Survey and an Empirical Analysis. *Journal of Engineering Research*. https://doi.org/10.1016/j.jer.2024.02.018

Theodorakis, Y., Georgiadis, K., & Hassandra, M. (2024). Evolution of the Olympic Movement: Adapting to Contemporary Global Challenges. *Social Sciences*, *13*(7), 326. https://doi.org/10.3390/socsci13070326

Verghese, A., T, Sudalaimuthu., & S, Visalaxi. (2021). Analysis and Forecasting Covid-19 Spread in India Using Logistic Regression and Prophet Time Series. *2021 International Conference on Computational Performance Evaluation (ComPE)*, 928–932. https://doi.org/10.1109/ComPE53109.2021.9752218

Wen, J., & Wang, X. (2020). Study of the Visualization and Interaction of Data: Take the Historical Data of the Winter Olympics as an Example. *2020 International Conference on Innovation Design and Digital Technology (ICIDDT)*, 78–82. https://doi.org/10.1109/ICIDDT52279.2020.00022

Wu, P., Zhu, X., Yang, S., & Huang, J. (2023). The Influence of the Beijing Winter Olympic Games on the Demand for Winter Sports: An Empirical Analysis Based on the Baidu Index. *Heliyon*, *9*(10), e20426. https://doi.org/10.1016/j.heliyon.2023.e20426

Wulandari, R., Surarso, B., Irawanto, B., & Farikhin, F. (2021). The Forecasting of Palm Oil Based on Fuzzy Time Series-Two Factor. *Journal of Soft Computing Exploration*, *2*(1), 11–16. https://shmpublisher.com/index.php/joscex/article/view/14

Xinyi, S., & Chenglong, X. (2022). Visual Analysis of the Distribution Characteristics and Influencing Factors of Olympic Medals. *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 1633–1637. https://doi.org/10.1109/IMCEC55388.2022.10019867

Yin, K., Guo, H., & Yang, W. (2024). A Novel Real-Time Multi-Step Forecasting System with a Three-Stage Data Preprocessing Strategy for Containerized Freight Market. *Expert Systems with Applications*, *246*, 123141. https://doi.org/10.1016/j.eswa.2024.123141

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 33 – 47
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Performance Evaluation of Long Short-Term Memory for Chili Price Prediction

Fata Nabil Fikri [(1)*], Nurochman [(2)]
Department of Informatics, Universitas Islam Negeri Sunan Kalijaga, Yogyakarta, Indonesia
e-mail : fatanabil88@gmail.com, nurochman@uin-suka.ac.id.
* Corresponding author.
This article was submitted on 23 February 2024, revised on 8 April 2024, accepted on 15 April 2024, and published on 31 January 2025.

Abstract

Groceries prices often experience fluctuations in several regions in Indonesia, such as East Java Province and one of the commodities is chilies, both red chilies and rawit chilies. Predictive steps that utilize machine learning such as Long-Short Term Memory (LSTM) can be taken to estimate the next price of chili with expectations that the appropriate strategy can be taken by the authorities. LSTM is a network that developed from RNN networks in previous times by offering a longer cell memory so that more information can be stored. This research focuses on finding out whether the LSTM network can be applied to the case of chili price prediction and what architecture and hyperparameter configuration is appropriate for this case. For this reason, the experimental method is used by testing several predetermined variables to obtain the right architecture and hyperparameter configuration. The results of this research show that the LSTM network can be applied in this case and the architecture and best hyperparameter configuration obtained are the same for both types of chilies, namely red chilies and rawit chilies. For red chili, the best RMSE value that can be produced is 1751.890 and 1888.741 for rawit chili.

Keywords: LSTM, Prediction, RMSE, Chili Prices, Groceries

Abstrak

Harga bahan pangan sering terjadi fluktuasi di beberapa daerah di Indonesia seperti di Provinsi Jawa Timur dan salah satu komoditasnya yaitu cabai, baik cabai merah maupun cabai rawit. Langkah prediksi yang memanfaatkan pembelajaran mesin seperti Long-short Term Memory (LSTM) dapat ditempuh untuk memperkirakan harga cabai selanjutnya dengan harapan strategi yang tepat dapat diambil oleh pihak yang berwenang. LSTM merupakan bentuk jaringan hasil pengembangan dari jaringan RNN pada masa-masa sebelumnya dengan menawarkan memori sel yang lebih panjang sehingga lebih banyak infromasi yang dapat disimpan. Penelitian ini berfokus untuk mengetahui apakah jaringan LSTM dapat diterapkan pada kasus prediksi harga cabai serta arsitektur dan konfigurasi hyperparameter apa yang tepat untuk kasus ini. Untuk itu, metode eksperimen ditempuh dengan menguji beberapa variabel yang telah ditentukan untuk mendapatkan arsitektur serta konfigurasi hyperparameter yang tepat. Hasil dari penelitian ini menunjukkan bahwa jaringan LSTM dapat diterapkan pada kasus ini dan arsitektur serta konfigurasi hyperparameter terbaik yang didapat itu sama untuk kedua jenis cabai yaitu cabai merah dan cabai rawit. Pada data cabai merah nilai RMSE terbaik yang dapat dihasilkan yaitu 1751,890 dan 1888,741 pada data cabai rawit.

Kata Kunci: LSTM, Prediksi, RMSE, Harga Cabai, Bahan Pangan

1. INTRODUCTION

The demand for basic commodities in Indonesia is highly significant for its population. Particularly, the prices of these basic commodities frequently experience instability or fluctuation, as is evident with chili, a key commodity. Yanwardhana, reporting in CNBC Indonesia, noted that the average price of chili in Indonesia can reach Rp 106,764 per kilogram, with prices even exceeding Rp 150,000 in some regions (Yanwardhana, 2022). In East Java Province, the price of red chili soared by 241.48%, increasing from Rp 24,840 per kilogram to Rp 84,823 per kilogram as of June 7, 2022. A similar increase was observed in red chili prices, which rose approximately 78.58%. Such instability can be attributed to several factors, including weather conditions, fuel prices, and

significant festive seasons. As quoted from CNN Indonesia, the price of chili rose ahead of the Ramadan period from Rp 55,000 per kilogram to Rp 60,000 per kilogram.

With advancements in computational technology, predictive models leveraging machine learning can be developed. One widely used model for prediction and forecasting is the Long Short-Term Memory (LSTM). LSTM is a type of artificial neural network categorized under Recurrent Neural Networks (RNNs). RNNs are capable of addressing time-series data problems due to their inherent “memory” within their cells. However, RNNs face limitations such as the vanishing gradient problem, which slows training progress. LSTM, introduced in the 1990s, was developed to address this challenge by providing longer memory capabilities (Yadav et al., 2020).

Hochreiter and Schmidhuber in their study highlighted that LSTM resolves issues arising from Back-Propagation Through Time (BPTT) and Real-Time Recurrent Learning (RTRL) algorithms, where error signals in these algorithms either blow up or vanish as they propagate backward in time. The former causes weight oscillation, while the latter extends model training duration significantly. LSTM is specifically designed to handle such errors, enabling the model to learn data spanning over 1,000 steps (Hochreiter & Schmidhuber, 1997).

Several past studies have employed machine learning for prediction purposes. For instance, Chairurrachman I conducted research on PT Indofood CBP Sukses Makmur Tbk stock prices using the LSTM network. The study revealed that CNN-LSTM architecture achieved the best MAE value of 74.1365 (Chairurrachman, 2022). Additionally, research by Arfan and Lussiana compared LSTM and Support Vector Regression (SVR). Using stock prices from various companies as the dataset, their findings demonstrated that LSTM produced better accuracy than SVR, particularly for longer time-series data (Arfan & ETP, 2020).

Similarly, Riyantoko et al. (2020) analyzed predictions for banking sector stock prices using the LSTM algorithm. This study examined the impact of varying epochs and optimization techniques on computation time, RMSE, and loss levels. It compared the optimizations Adam, RMSprop, and SGD, finding that Adam and RMSprop achieved similar accuracy levels ranging from 89% to 95%, while SGD lagged behind at 49% to 61%. Additionally, changing the number of epochs affected computation time but had little impact on the resulting RMSE. Another relevant study by Syaidah et al. (2020) focused on predicting staple food prices in Jakarta using Artificial Neural Networks (ANN). Their results indicated that the chosen alpha and threshold values influenced accuracy, with lower values enhancing accuracy.

Further, a study conducted by Suradiradja (2022), titled “Machine Learning Algorithms: Multi-Layer Perceptron and Recurrent Neural Network for Predicting Large Red Chili Prices in Tangerang City,” showed notable results. The study achieved a low MAPE value of 3.79%, indicating significant accuracy, using the Multi-Layer Perceptron algorithm. However, it compared only two algorithms: Recurrent Neural Networks and Multi-Layer Perceptron. Currently, LSTM algorithms are frequently used in research for time-series data predictions, owing to their superior accuracy compared to earlier algorithms. The primary focus of this study is to implement LSTM, determine its architecture, and optimize several hyperparameters for effective data preparation and training for predicting chili prices in East Java Province.

2. METHODS

2.1 Long Short-Term Memory (LSTM)

LSTM, or Long Short-Term Memory, is a type of artificial neural network architecture commonly applied in cases involving time-series data, text, video, or audio. LSTM represents a significant advancement over Recurrent Neural Networks (RNNs), which are ineffective at handling long-term dependencies due to their lack of persistent "memory" cells. As a result, LSTM outperforms RNNs in such scenarios. Figure 1 illustrates the structure of an LSTM cell.

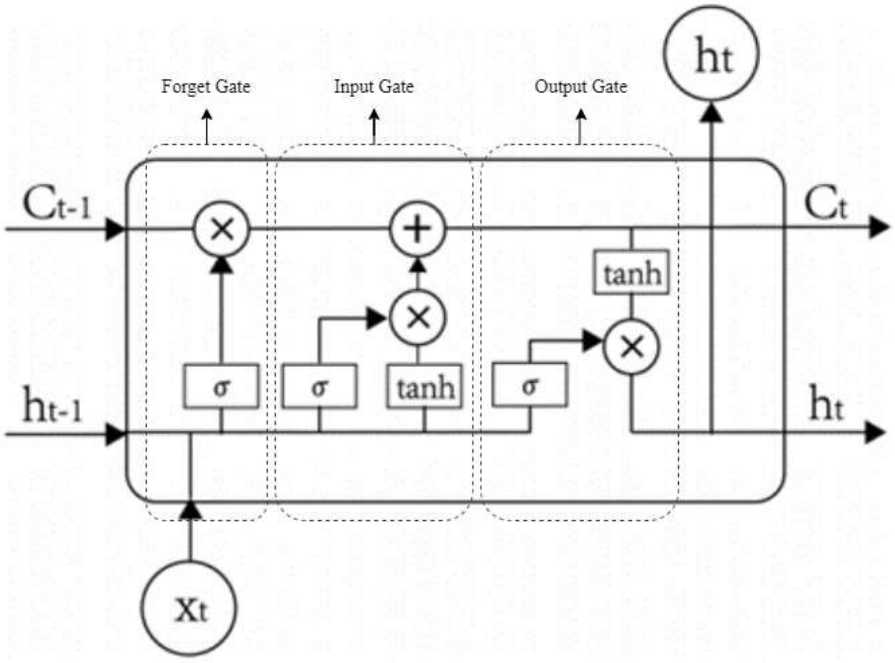


Figure 1 LSTM Structure (Qiu et al., 2020)

Based on Figure 1, there are several components, often referred to as "gates," within an LSTM cell (Istiake Sunny et al., 2020). A gate, represented by a dashed box labeled "Forget Gate," determines whether the information from the previous state should be retained or discarded. The forget gate computes the hidden state from the previous step and the current input state using a Sigmoid activation function.

Another gate, also represented by a dashed box labeled "Input Gate," evaluates whether newly incoming information is important. If deemed important, the information is added to the current state; otherwise, it is discarded. The input gate involves two computations: the input gate value using the Sigmoid activation function and the memory cell candidate value using the Tanh activation function. An output gate, shown as a dashed box labeled "Output Gate," determines the output value based on the processed information from the forget and input gates. The value of the output gate is computed, and the result becomes the value for the next hidden state.

2.2 Tools

The tools utilized in this study include both software and hardware, described as follows. Python is frequently used in machine learning, data analysis, and various aspects of computer science. It can execute without the need for compilation, unlike many other programming languages. Additionally, Python offers a package manager known as PIP, which provides access to numerous libraries related to computer science. TensorFlow is one of the most renowned libraries in the machine learning community. It is used for building computational models applicable in pattern recognition, image recognition, prediction, and more. TensorFlow is popular due to its multi-level abstraction, flexible coding, and comprehensive ecosystem (Joseph et al., 2021).

Keras API is one limitation of TensorFlow is its steep learning curve for beginners. To address this, the Keras API was developed to simplify building, training, and evaluating computational models. Keras is built on top of TensorFlow and focuses on user-friendly machine learning implementations (Amendolara et al., 2023). The hardware used includes a notebook equipped with a GPU (Graphics Processing Unit). Machine learning benefits from the GPU's Compute Unified Device Architecture (CUDA) to enhance computational performance. CUDA is a parallel computing platform developed by NVIDIA.

2.3 Experimental Methodology

The experimental method involves investigating relationships between variables. Variables in the experimental methodology are categorized as follows:

a) **Independent variables:** These are manipulated to observe the response of dependent variables under specific scenarios.
b) **Control variables:** These remain constant throughout the experiment to neutralize their impact on dependent variables.
c) **Dependent variables:** These are observed to understand the effect of changes in independent variables.

The flow diagram illustrating the experimental stages in this research is presented in Figure 2. The steps are as follows. For data collection and preparation, the dataset for training is collected and prepared as input for the LSTM model. This process involves several stages such as cleaning, filling in missing data, normalizing using MinMaxScaling, and splitting the dataset into training and testing data. After that experiment designs stage ensures that the experiment is conducted consistently and with focus, determining the variables relevant to the execution of the experiment.

The experiment is carried out according to the scenarios planned in the previous stage. The resulting experimental data is recorded for later analysis. The recorded data is analyzed to identify which scenario produces the best results based on evaluation metrics. Additionally, this step explains the findings of the conducted experiments. Conclusions are drawn to succinctly summarize the experimental results, making them easier to comprehend and addressing the problem statements identified.

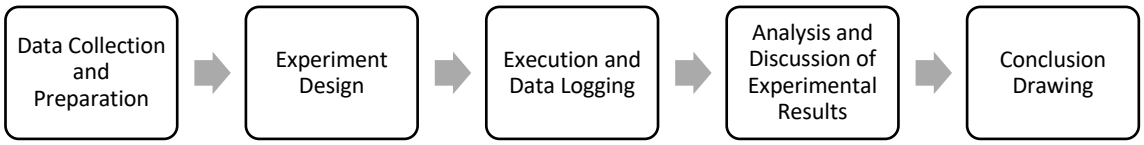


Figure 2 Experimental Workflow Diagram

2.4 Data collection and preparation

The dataset used in this study consists of chili price data from East Java Province over approximately three years (January 1, 2020 – June 1, 2023). It includes two types of commodities: red chili and bird's eye chili. This price data was obtained from the PIHPS Nasional website (National Strategic Food Price Information Center), managed by Bank Indonesia (Rahmadini et al., 2023). The dataset is presented in a .xlsx file format containing daily chili prices in East Java. Figure 3 shows the price trends of red chili and bird's eye chili in East Java from January 1, 2020, to June 1, 2023.

The blue line in the graph represents red chili prices, while the orange line represents bird's eye chili prices. The x-axis indicates the sequence of dates from the start to the end of data collection, and the y-axis represents the actual prices of red and bird's eye chili on their respective dates. The total chili price data collected for the specified date range amounted to 894 records. However, this figure includes invalid data, such as missing or non-numeric values (e.g., a "-" character). Therefore, data cleaning was conducted to eliminate such anomalies, resulting in 851 valid data entries.

The next preparation step involved filling or replacing missing or undesirable data points. This step is crucial as predictive processes require uniform data intervals. For example, if the interval between data points is one day, all intervals should consistently adhere to this pattern. In this dataset, missing entries caused inconsistencies in intervals, either due to unwanted data values

or entirely missing values on specific dates. For instance, no reports were available on Saturdays and Sundays because the PIHPS Nasional website does not update data on these days.

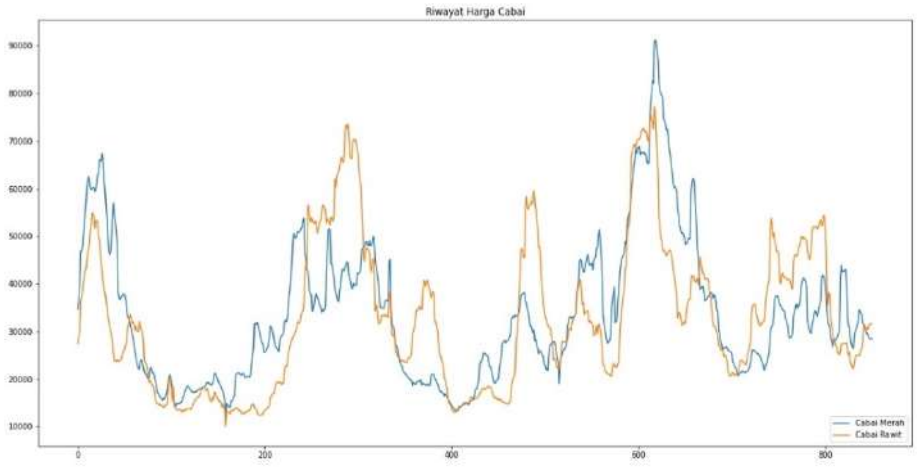


Figure 3 Red Chili and Bird's Eye Chili Price Trends

Several methods can be applied to address missing data, including using the value from the previous date, calculating the average of the values before and after the missing data, or using the value from the next date, among others. In this study, missing data was filled using the value from the preceding day. This method was chosen because the website updates data daily. Using the average of preceding and succeeding values would be impractical as succeeding values were not always available at the time of calculation. This data-filling process resulted in 1,250 values, which were then further processed.

After data cleaning and filling missing values, the next preparation step was data normalization. Normalization aims to optimize the model's training process. A commonly used method for time-series dataset normalization is Min-Max Scaling. This method adjusts the data range to a specific interval, typically between 0 and 1, though other ranges can also be used if necessary (Deepa & Ramesh, 2022). The Min-Max Scaling equation is shown in Equation 1.

$$y = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

In this equation, $\boldsymbol{y}$ represents the normalized value within the desired range, typically from 0 to 1. The variable $\boldsymbol{x}$ is the original value from the dataset, while $\boldsymbol{x_{min}}$ and $\boldsymbol{x_{max}}$ are the minimum and maximum values in the dataset, serving as the lower and upper bounds, respectively. By transforming all data points using this formula, the values are rescaled proportionally within the specified range, optimizing the dataset for machine learning algorithms by reducing the risk of bias due to varying magnitudes in data points. This process ensures more uniform data distribution, simplifying the model's task of processing data during training.

The next step involved splitting the dataset into training and testing data. The split is typically based on a specific ratio between training and testing datasets. A common ratio used is 80:20, where 80% of the data is used to train the model, while the remaining 20% is reserved for testing its performance (Adinugroho, 2023). With a total of 1,250 data points, this ratio yielded 1,000 entries for training and 250 entries for testing. However, in this study, the training-to-testing ratio was treated as an independent variable to be evaluated and discussed further in subsequent sections. After splitting the data into training and testing sets, the training data was further processed into input-label pairs. Each input consists of a sequence of values of a certain length, referred to as the sequence length. The label generally consists of a single value.

For example, if the sequence length is 7, the values from Day 1 to Day 7 form one sequence or input, while the label is the value on Day 8. This sequence formation process shifts sequentially, with each new sequence beginning at the next data point in the dataset. For instance, the second sequence consists of values from Day 2 to Day 8, with the label being the value on Day 9, and so on until all training data has been processed. However, in this study, the sequence length was not fixed as it was considered an independent variable to be further examined. An example of the final format of the sequence and label formation for training data is shown in Figure 4.

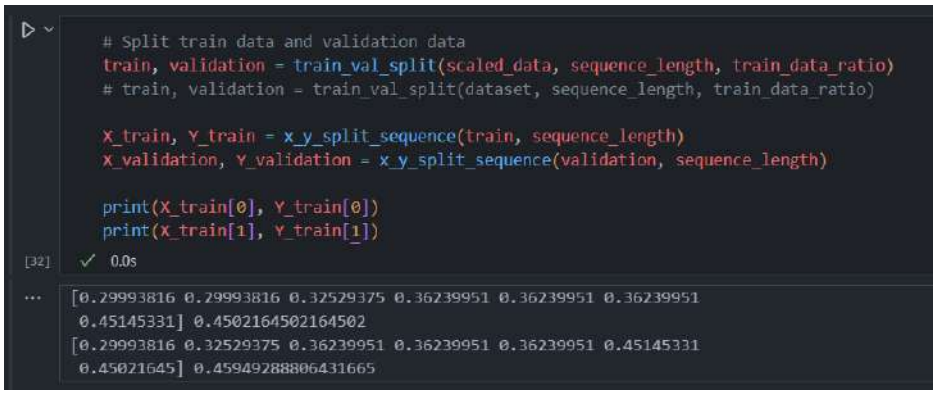


Figure 4 Example of Sequence Formation and Labeled Data from Training Data

2.5 Experimental Scenario Design

The design of experimental scenarios aims to achieve valid, relevant, and accountable results. Additionally, this design simplifies the experimental process and ensures that it remains controlled. In this study, seven scenarios were designed, applied to both types of chili data, namely red chili and bird's eye chili. The first three scenarios focus on LSTM architecture, as detailed in Tables 1, 2, and 3, while the remaining four scenarios relate to data, training, and model optimization, as explained in Tables 4, 5, 6, and 7.

Table 1 Scenario 1 (Number of Units)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	10	0	Linear	80:20	30	30	Adam
2	20	0	Linear	80:20	30	30	Adam
3	30	0	Linear	80:20	30	30	Adam
4	40	0	Linear	80:20	30	30	Adam
5	50	0	Linear	80:20	30	30	Adam
6	60	0	Linear	80:20	30	30	Adam
7	70	0	Linear	80:20	30	30	Adam
8	80	0	Linear	80:20	30	30	Adam
9	90	0	Linear	80:20	30	30	Adam
10	100	0	Linear	80:20	30	30	Adam

Table 2 Scenario 2 (Hidden Layers)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	80:20	30	30	Adam
2	50	1	Linear	80:20	30	30	Adam
3	50	2	Linear	80:20	30	30	Adam
4	50	3	Linear	80:20	30	30	Adam

Table 3 Scenario 3 (Activation Function)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	80:20	30	30	Adam
2	50	0	Relu	80:20	30	30	Adam
3	50	0	Leaky Relu	80:20	30	30	Adam
4	50	0	SELU	80:20	30	30	Adam

Table 4 Scenario 4 (Training Data Ratio)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	70:30	30	30	Adam
2	50	0	Linear	80:20	30	30	Adam
3	50	0	Linear	90:10	30	30	Adam

Table 5 Scenario 5 (Sequence Length)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	80:20	7	30	Adam
2	50	0	Linear	80:20	14	30	Adam
3	50	0	Linear	80:20	21	30	Adam
4	50	0	Linear	80:20	30	30	Adam

Table 6 Scenario 6 (Epoch)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	80:20	30	10	Adam
2	50	0	Linear	80:20	30	20	Adam
3	50	0	Linear	80:20	30	30	Adam
4	50	0	Linear	80:20	30	40	Adam
5	50	0	Linear	80:20	30	50	Adam

Table 7 Scenario 7 (Model Optimization)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	80:20	30	30	Adam
2	50	0	Linear	80:20	30	30	RMSprop
3	50	0	Linear	80:20	30	30	SGD

Scenario 1 evaluates the number of LSTM units (independent variable) with varying values, while other variables are treated as control variables, kept constant across all trials in this scenario. **Scenario 2** examines the number of hidden layers between the input and output layers, testing variations of 1, 2, and 3 layers, as well as a condition without hidden layers. Other variables remain as controls. **Scenario 3** focuses on testing different activation functions applied to the output layer, including Linear (Dubey et al., 2022), Relu, Leaky Relu (Gustineli, 2022), and SELU.

Scenario 4 investigates aspects of data, training, and model optimization by treating the training-to-testing ratio as an independent variable with several values tested. **Scenario 5** evaluates the

sequence length with various values such as 7, 14, 21, and 30, representing time periods of 1 week, 2 weeks, 3 weeks, and 1 month, respectively (Li et al., 2021). **Scenario 6** tests the number of epochs run during training to observe its effect on model performance (Moghar & Hamiche, 2020). The final scenario, **Scenario 7**, tests various model optimization methods, including Adam, RMSprop, and SGD, which are commonly used in machine learning to enhance model performance (Irfan et al., 2023).

2.6 Evaluation metrics

Evaluation metrics are measures used to assess the outcomes of an experiment. In this study, a key metric is the accuracy level of a model. Additionally, the training duration of a model is considered to evaluate performance. However, accuracy has greater significance compared to training duration, which is only considered further when models yield similar accuracy levels. These evaluation metrics also serve as dependent variables in this experiment. Below are the metrics used in this study. The Root Mean Squared Error (RMSE) equation measures the discrepancy between the predicted values and actual values in a dataset. It is calculated by first squaring the differences between each actual value (y_i) and its corresponding predicted value ($\hat{y}_i$), then summing these squared differences across all data points. This total is divided by the number of data points (n) to calculate the mean squared error, and finally, the square root is taken to obtain the RMSE. A smaller RMSE value indicates better model performance, as it reflects a lower prediction error.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (2)$$

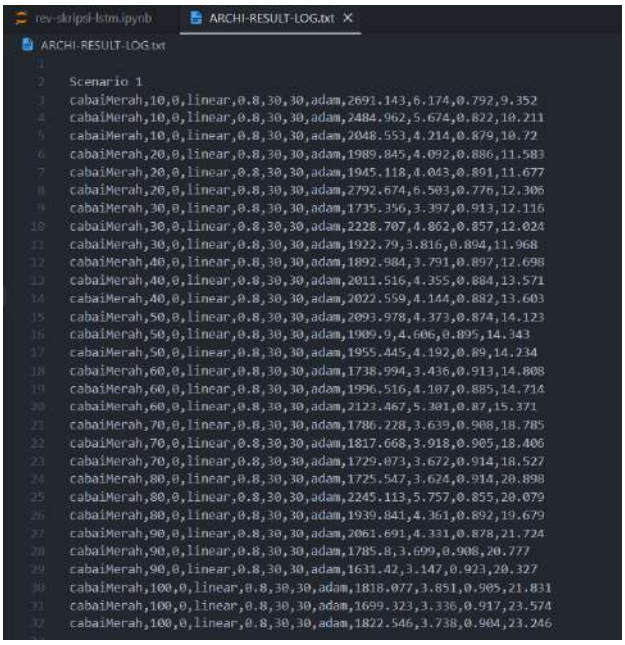
The Mean Absolute Error Percentage (MAPE) equation measures prediction error as a percentage, making it easier to interpret across datasets of different scales. For each data point, the absolute difference between the actual value (y_i) and the predicted value ($\hat{y}_i$) is calculated, divided by the actual value (y_i), and multiplied by 100 to convert it into a percentage. These percentages are then averaged across all data points (n) to yield the MAPE. Lower MAPE values indicate higher model accuracy and better prediction performance.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \cdot 100 \quad (3)$$

The final evaluation metric used in this experiment is training duration. It refers to the time taken for the LSTM model to complete training, from the first epoch to the last. Training duration is measured in seconds, starting right before the training process begins and ending upon its completion.

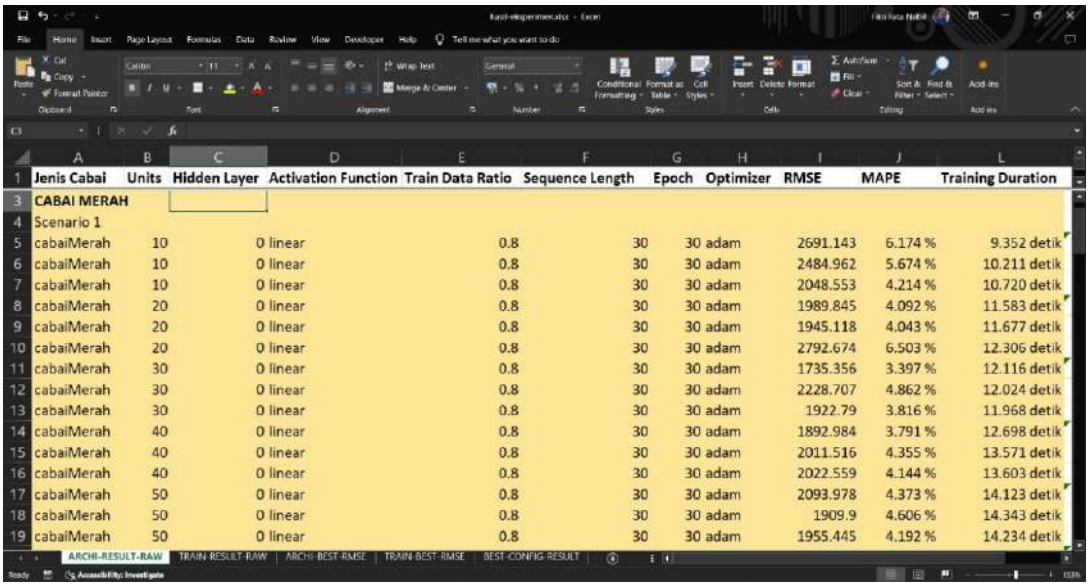
2.7 Experiment Execution and Results Recording

The experiment was conducted by running all scenarios sequentially, from the initial to the final scenario, for both red chili and bird's eye chili data. Each independent variable value was tested three times, and the results of each test were recorded. The average RMSE value from the three tests was then calculated, and the lowest RMSE value was selected as the basis for comparing scenarios. The experiment results were automatically recorded each time a scenario was executed, saved in a .txt file format with columns separated by commas (","). This results recording process is illustrated in Figure 5. To facilitate the readability of the results, the data was transformed into a tabular format using spreadsheet software (Microsoft Excel) and appropriate column headers were added. Figure 6 shows the experimental results in tabular format.



```
Scenario 1
cabaiMerah,10,0,linear,0.8,30,30,adam,2691.143,6.174,0.792,9.352
cabaiMerah,10,0,linear,0.8,30,30,adam,2484.962,5.674,0.822,10.211
cabaiMerah,10,0,linear,0.8,30,30,adam,2048.553,4.214,0.879,10.72
cabaiMerah,20,0,linear,0.8,30,30,adam,1989.845,4.092,0.886,11.583
cabaiMerah,20,0,linear,0.8,30,30,adam,1945.118,4.043,0.891,11.677
cabaiMerah,20,0,linear,0.8,30,30,adam,2792.674,6.503,0.776,12.306
cabaiMerah,30,0,linear,0.8,30,30,adam,1735.356,3.397,0.913,12.116
cabaiMerah,30,0,linear,0.8,30,30,adam,2228.707,4.862,0.857,12.024
cabaiMerah,30,0,linear,0.8,30,30,adam,1922.79,3.816,0.894,11.968
cabaiMerah,40,0,linear,0.8,30,30,adam,1892.984,3.791,0.897,12.698
cabaiMerah,40,0,linear,0.8,30,30,adam,2011.516,4.355,0.884,13.571
cabaiMerah,40,0,linear,0.8,30,30,adam,2022.559,4.144,0.882,13.603
cabaiMerah,50,0,linear,0.8,30,30,adam,2093.978,4.373,0.874,14.123
cabaiMerah,50,0,linear,0.8,30,30,adam,1909.9,4.606,0.895,14.343
cabaiMerah,50,0,linear,0.8,30,30,adam,1955.445,4.192,0.89,14.234
cabaiMerah,60,0,linear,0.8,30,30,adam,1738.994,3.436,0.913,14.808
cabaiMerah,60,0,linear,0.8,30,30,adam,1996.516,4.107,0.885,14.714
cabaiMerah,60,0,linear,0.8,30,30,adam,2123.467,5.301,0.87,15.371
cabaiMerah,70,0,linear,0.8,30,30,adam,1786.228,3.639,0.908,18.785
cabaiMerah,70,0,linear,0.8,30,30,adam,1817.668,3.918,0.905,18.406
cabaiMerah,70,0,linear,0.8,30,30,adam,1729.073,3.672,0.914,18.527
cabaiMerah,80,0,linear,0.8,30,30,adam,1725.547,3.624,0.914,20.898
cabaiMerah,80,0,linear,0.8,30,30,adam,2245.113,5.757,0.855,20.079
cabaiMerah,80,0,linear,0.8,30,30,adam,1939.841,4.361,0.892,19.679
cabaiMerah,90,0,linear,0.8,30,30,adam,2061.691,4.331,0.878,21.724
cabaiMerah,90,0,linear,0.8,30,30,adam,1785.8,3.699,0.908,20.777
cabaiMerah,90,0,linear,0.8,30,30,adam,1631.42,3.147,0.923,20.327
cabaiMerah,100,0,linear,0.8,30,30,adam,1818.077,3.851,0.905,21.831
cabaiMerah,100,0,linear,0.8,30,30,adam,1699.323,3.336,0.917,23.574
cabaiMerah,100,0,linear,0.8,30,30,adam,1822.546,3.738,0.904,23.246
```

Figure 5 Experimental Results Recording



Jenis Cabai	Units	Hidden Layer	Activation Function	Train Data Ratio	Sequence Length	Epoch	Optimizer	RMSE	MAPE	Training Duration
CABAI MERAH										
Scenario 1										
cabaiMerah	10	0	linear	0.8	30	30	adam	2691.143	6.174 %	9.352 detik
cabaiMerah	10	0	linear	0.8	30	30	adam	2484.962	5.674 %	10.211 detik
cabaiMerah	10	0	linear	0.8	30	30	adam	2048.553	4.214 %	10.720 detik
cabaiMerah	20	0	linear	0.8	30	30	adam	1989.845	4.092 %	11.583 detik
cabaiMerah	20	0	linear	0.8	30	30	adam	1945.118	4.043 %	11.677 detik
cabaiMerah	20	0	linear	0.8	30	30	adam	2792.674	6.503 %	12.306 detik
cabaiMerah	30	0	linear	0.8	30	30	adam	1735.356	3.397 %	12.116 detik
cabaiMerah	30	0	linear	0.8	30	30	adam	2228.707	4.862 %	12.024 detik
cabaiMerah	30	0	linear	0.8	30	30	adam	1922.79	3.816 %	11.968 detik
cabaiMerah	40	0	linear	0.8	30	30	adam	1892.984	3.791 %	12.698 detik
cabaiMerah	40	0	linear	0.8	30	30	adam	2011.516	4.355 %	13.571 detik
cabaiMerah	40	0	linear	0.8	30	30	adam	2022.559	4.144 %	13.603 detik
cabaiMerah	50	0	linear	0.8	30	30	adam	2093.978	4.373 %	14.123 detik
cabaiMerah	50	0	linear	0.8	30	30	adam	1909.9	4.606 %	14.343 detik
cabaiMerah	50	0	linear	0.8	30	30	adam	1955.445	4.192 %	14.234 detik

Figure 6 Experimental Results in Tabular Format

3. RESULTS AND DISCUSSION

After running all scenarios, the best results (lowest averages) for each scenario, for both red chili and bird's eye chili, are presented. The best results for red chili are shown in Table 8. The RMSE values across all scenarios range from 1600 to 1900, with the lowest RMSE observed in Scenario 7 during RMSprop model optimization, resulting in an RMSE of 1635.065. Regarding MAPE, all scenarios achieved values below 4%, with the smallest MAPE also found in Scenario 7 during RMSprop optimization.

The longest training duration occurred in Scenario 6 when testing the number of epochs, as it used the highest value of 50 epochs. However, the smallest average RMSE was observed in Scenario 6, at 1754.700. Conversely, Scenario 7, which had the lowest RMSE value, produced a

higher average RMSE compared to Scenario 6. Table 9 compares the average RMSE values in Scenarios 6 and 7. In Scenario 7, the results of RMSprop optimization tests were inconsistent, as not all tests achieved low RMSE values (close to the minimum RMSE). As a result, the average RMSE in Scenario 7 was relatively high. On the other hand, the RMSE values in Scenario 6 were more consistent, staying within the 1700 range, and thus yielded a lower average RMSE than Scenario 7.

Table 8 Best Results for Red Chili Price Prediction

No.	Scenario	Independent Variable	Variable Value	RMSE	MAPE (%)	Training Duration (s)	Average RMSE
1	Scenario 1	Number of LSTM Units	70	1729.073	3.67%	18.527	1777.656
2	Scenario 2	Hidden Layers	0	1663.543	3.21%	13.744	1781.956
3	Scenario 3	Activation Function	Linear	1861.336	3.90%	14.337	1953.674
4	Scenario 4	Training Data Ratio	80:20	1710.004	3.52%	10.886	1975.861
5	Scenario 5	Sequence Length	7	1932.669	3.93%	5.431	1923.455
6	Scenario 6	Epochs	50	1739.348	3.52%	19.127	1754.700
7	Scenario 7	Optimization Method	RMSprop	1635.065	3.16%	14.777	1931.978

Table 9 Comparison of Average RMSE Between Scenario 6 and Scenario 7 for Red Chili Price Prediction

No.	Scenario	Independent Variable	RMSE 1	RMSE 2	RMSE 3	Average RMSE
1	Scenario 6	50 Epochs	1778.592	1746.161	1739.348	1754.700
2	Scenario 7	RMSprop Optimization	1987.332	1635.065	2173.538	1931.978

Table 10 Best Results for Bird's Eye Chili

No.	Scenario	Independent Variable	Independent Variable Value	RMSE	MAPE	Training Duration	Average RMSE
1	Scenario 1	Number of Units	80	1913.818	3.59 %	19.936 seconds	1969.866
2	Scenario 2	Hidden Layers	0	1929.001	3.55 %	11.222 seconds	2099.155
3	Scenario 3	Activation Function	Linear	2056.248	4.01 %	14.694 seconds	2104.328
4	Scenario 4	Training Data Ratio	90:10	1941.933	3.384 %	15.183 seconds	2067.900
5	Scenario 5	Sequence Length	7	1936.939	3.485 %	6.531 seconds	2061.045
6	Scenario 6	Epochs	50	1837.712	3.419 %	22.351 seconds	1878.329
7	Scenario 7	Model Optimization	RMSprop	1834.365	3.423 %	15.001 seconds	1983.437

Table 10 presents the best results for bird's eye chili in each scenario. Across all scenarios, RMSE values for bird's eye chili ranged higher than those for red chili, with values between 1800 and 2000. The lowest RMSE occurred in Scenario 7 during RMSprop optimization testing. The smallest MAPE, however, was achieved in Scenario 4, during the 90:10 training-to-test data ratio test, with a MAPE of 3.384. This indicates that MAPE does not always correlate with RMSE values.

The training durations required for bird's eye chili were similar to those for red chili. The longest duration for bird's eye chili occurred in the 50-epoch test in Scenario 6, lasting 22.351 seconds. The training duration difference between the two datasets was only around 3 seconds, suggesting similar training time requirements for both data types. As with red chili, scenarios with the smallest RMSE values did not always result in the smallest average RMSE for bird's eye chili. In Scenario 7, RMSprop optimization testing produced the lowest RMSE (1834.365), but the smallest average RMSE was observed in Scenario 6, during the 50-epoch test. The inconsistency of RMSE values in Scenario 7 contributed to a higher average RMSE.

Table 11 Comparison of Average RMSE Between Scenario 6 and Scenario 7 for Bird's Eye Chili Price Prediction

No.	Scenario	Independent Variable	RMSE 1	RMSE 2	RMSE 3	Average RMSE
1	Scenario 1	Number of Units	80	1913.818	3.59 %	19.936 seconds
2	Scenario 2	Hidden Layers	0	1929.001	3.55 %	11.222 seconds

Table 11 shows that the RMSE range from RMSprop optimization testing is quite broad, with a minimum value of 1834.365 and a maximum of 2174.004. In contrast, the RMSE range during the 50-epoch test was narrower, with a minimum of 1837.712 and a maximum of 1974.733. Based on this discussion, the scenario with the smallest average RMSE is Scenario 6 during the 50-epoch test, for both red chili and bird's eye chili. This variable configuration is shown in Table 6. Scenario 6 with 50 epochs was retested five times to confirm it as the best scenario. The retesting was performed for both chili types, and the results are shown in Table 12.

Table 12 Re-Test Results for Scenario 6 Epoch 50

No.	Type of Chili	RMSE	MAPE	Training Duration	Average RMSE
1	Red Chili (Cabai Merah)	1540.898	2.923 %	22.270 seconds	1751.690
2	Bird's Eye Chili (Cabai Rawit)	1816.318	3.503 %	28.906 seconds	1888.741

Table 12 indicates that the retest results for Scenario 6 with 50 epochs produced average RMSE values similar to those observed in the initial test, as seen in Tables 9 and 11. However, Table 12 provides enough evidence that Scenario 6 with 50 epochs yields more consistent RMSE values than the other scenarios. Figures 7 and 8 illustrate the retest results for Scenario 6 with 50 epochs for red chili and bird's eye chili. These show good accuracy levels, with a narrow gap between the prediction curve (orange) and the actual curve (blue). Although there are some wider gaps visible, as in Figure 6, the RMSE results in Table 12 show that red chili has a lower RMSE than bird's eye chili. For further testing, the trained model configured with Scenario 6 was tested on data outside the training dataset (price data for red chili and bird's eye chili from June 1, 2023, to October 18, 2023). After applying the same data preparation process used for the training dataset, 140 data points were obtained, and the model produced RMSE and MAPE values, as shown in Table 13.

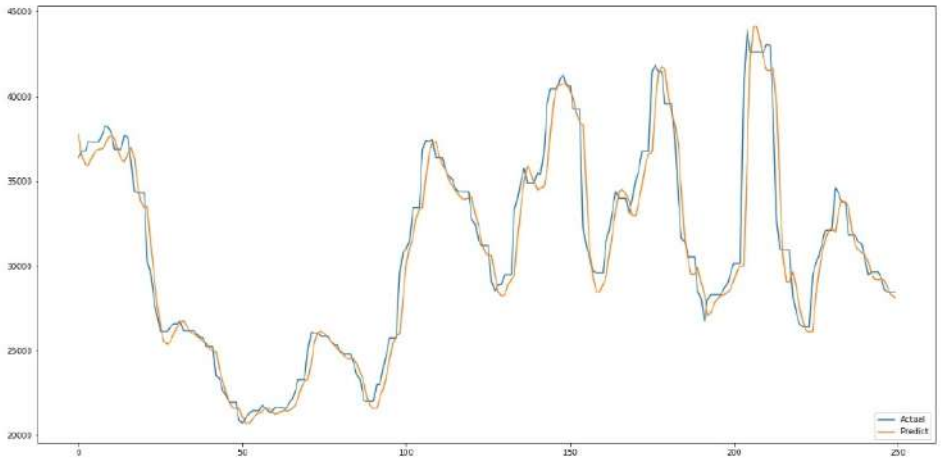


Figure 7 Graph of Re-Test Results for Scenario 6 Epoch 50 on Red Chili

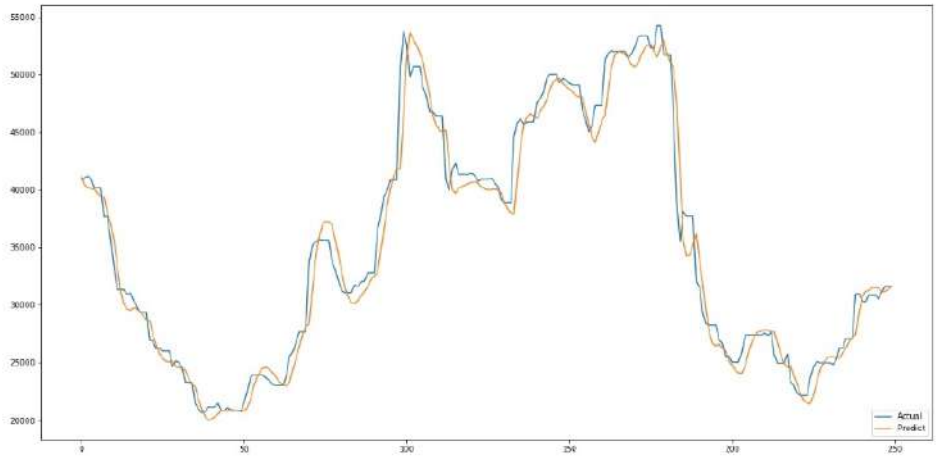


Figure 8 Graph of Re-Test Results for Scenario 6 Epoch 50 on Bird's Eye Chili

Both types of chili, red chili and bird's eye chili, produced very low RMSE values compared to those from the previous scenarios, as well as low MAPE values below 2.5%. This indicates that the trained model does not suffer from overfitting and generalizes the price patterns of both chili types well. Overfitting occurs when a model performs well on the training dataset but poorly on the test dataset because it focuses too much on the training details without recognizing general patterns in the test data (Montesinos López et al., 2022). In this case, the trained model did not overfit, as the RMSE values produced on the test data were not significantly different from those on the training data, as evidenced in Tables 12 and 13, which show test results on data beyond the training dataset range.

Table 13 Results of the Trained Model Testing

No.	Type of Chili	RMSE	MAPE
1	Red Chili (Cabai Merah)	1160.695	2.280 %
2	Bird's Eye Chili (Cabai Rawit)	816.052	2.256 %

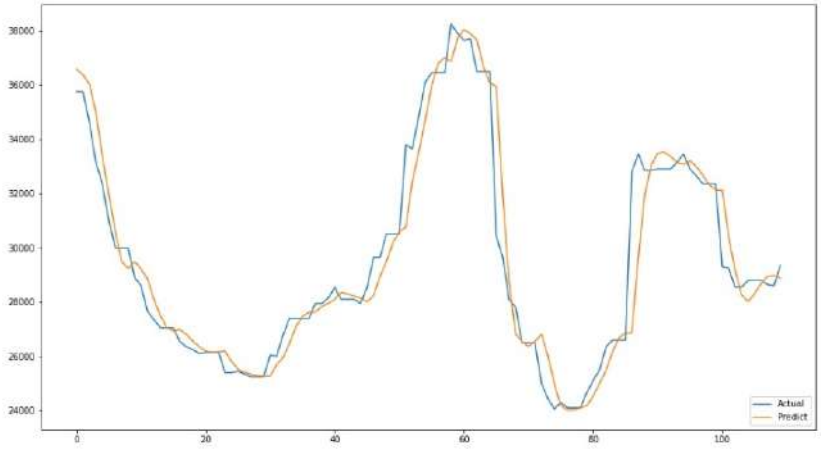


Figure 9 Grafik Hasil Pengujian Model yang Sudah Dilatih pada Data Cabai Merah

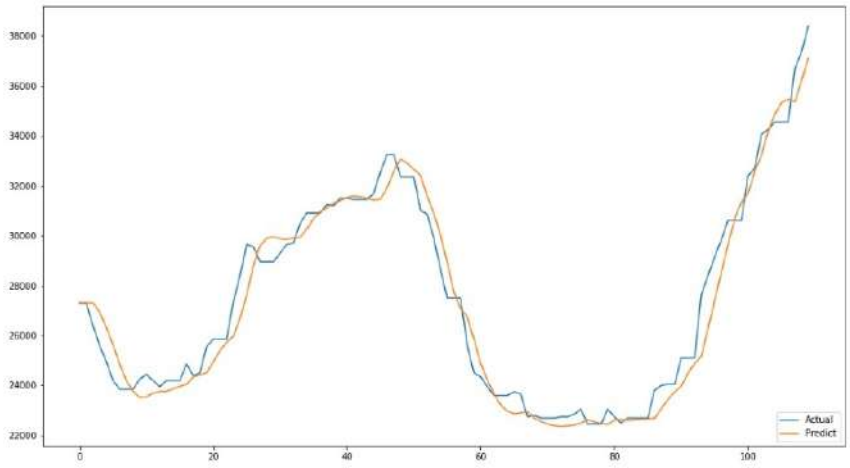


Figure 10 Grafik Hasil Pengujian Model yang Sudah Dilatih pada Data Cabai Rawit

Graphs depicting the test results on the external dataset for both chili types can be seen in Figures 9 and 10. The prediction curve (orange) and the actual curve (blue) are almost overlapping, with only slight gaps between the two. This demonstrates that the trained model produces highly accurate predictions when tested on a different dataset, indicating its strong ability to generalize patterns from unseen data.

4. CONCLUSIONS

In this study, the researchers attempted to predict chili prices in East Java Province using the Long Short-Term Memory (LSTM) network, a method in machine learning. To achieve this goal, the researchers designed and conducted experiments to determine the LSTM architecture and effective variable configurations for this case. The results show that the LSTM network is well-suited for chili price data, producing satisfactory results in predicting chili prices in East Java. Among all scenarios tested, Scenario 6, with 50 epochs, achieved the best results with the smallest RMSE and MAPE. The average RMSE values from the retest of Scenario 6 were 1751.690 for red chili and 1888.741 for bird's eye chili. These results suggest that the developed model has strong predictive capabilities and generalizes well for both chili types.

For future research, it is recommended to test the LSTM architecture on other types of time-series data, such as stock prices or weather data, to expand its application. Furthermore, it is advised

to develop an information system that implements this chili price prediction model, enabling stakeholders to plan chili pricing and distribution more efficiently. With these measures, the results of this study could provide broader and more impactful contributions to the field of agricultural commodity price prediction.

REFERENCES

Adinugroho, R. (2023). *Perbandingan Rasio Split Data Training dan Data Testing Menggunakan Metode LSTM Dalam Memprediksi Harga Indeks Saham Asia* [UIN Syarif Hidayatullah Jakarta]. https://repository.uinjkt.ac.id/dspace/handle/123456789/67314

Amendolara, A. B., Sant, D., Rotstein, H. G., & Fortune, E. (2023). LSTM-Based Recurrent Neural Network Provides Effective Short Term Flu Forecasting. *BMC Public Health*, *23*(1), 1788. https://doi.org/10.1186/s12889-023-16720-6

Arfan, A., & ETP, L. (2020). Perbandingan Algoritma Long Short-Term Memory dengan SVR pada Prediksi Harga Saham di Indonesia. *PETIR*, *13*(1), 33–43. https://doi.org/10.33322/petir.v13i1.858

Chairurrachman, I. (2022). *Penerapan Long Short-Term Memory pada Data Time Series untuk Prediksi Harga Saham PT. Indofood CBP Sukses Makmur Tbk (ICBP)* [UIN Sunan Kalijaga Yogyakarta]. https://digilib.uin-suka.ac.id/id/eprint/53738/

Deepa, B., & Ramesh, K. (2022). Epileptic Seizure Detection Using Deep Learning Through Min Max Scaler Normalization. *International Journal of Health Sciences*, *6*, 10981–10996. https://doi.org/10.53730/ijhs.v6nS1.7801

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark. *Neurocomputing*, *503*, 92–108. https://doi.org/10.1016/j.neucom.2022.06.111

Gustineli, M. (2022). A Survey on Recently Proposed Activation Functions for Deep Learning. *Engineering Archive (Engrvix)*. https://doi.org/10.31224/2245

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Irfan, D., Gunawan, T. S., & Wanayumini, W. (2023). Comparison of SGD, Rmsprop, and Adam Optimation in Animal Classification Using CNNs. *International Conference on Information Science and Technology Innovation (ICoSTEC)*, *2*(1), 45–51. https://doi.org/10.35842/icostec.v2i1.35

Istiake Sunny, Md. A., Maswood, M. M. S., & Alharbi, A. G. (2020). Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model. *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 87–92. https://doi.org/10.1109/NILES50944.2020.9257950

Joseph, F. J. J., Nonsiri, S., & Monsakul, A. (2021). Keras and TensorFlow: A Hands-On Experience. In *Advanced Deep Learning for Engineers and Scientists* (pp. 85–111). Springer International Publishing. https://doi.org/10.1007/978-3-030-66519-7_4

Li, W., Kiaghadi, A., & Dawson, C. (2021). Exploring the Best Sequence LSTM Modeling Architecture for Flood Prediction. *Neural Computing and Applications*, *33*(11), 5571–5580. https://doi.org/10.1007/s00521-020-05334-3

Moghar, A., & Hamiche, M. (2020). Stock Market Prediction Using LSTM Recurrent Neural Network. *Procedia Computer Science*, *170*, 1168–1173. https://doi.org/10.1016/j.procs.2020.03.049

Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Overfitting, Model Tuning, and Evaluation of Prediction Performance. In *Multivariate Statistical Machine Learning Methods for Genomic Prediction* (pp. 109–139). Springer International Publishing. https://doi.org/10.1007/978-3-030-89010-0_4

Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting Stock Prices with Long-Short Term Memory Neural Network Based on Attention Mechanism. *PLOS ONE*, *15*(1), e0227222. https://doi.org/10.1371/journal.pone.0227222

Rahmadini, R., LorencisLubis, E. E., Priansyah, A., N, Y. R. W., & Meutia, T. (2023). Penerapan Data Mining untuk Memprediksi Harga Bahan Pangan di Indonesia Menggunakan Algoritma K-Nearest Neighbor. *Jurnal Mahasiswa Akuntansi Samudra*, *4*(4), 223–235. https://doi.org/10.33059/jmas.v4i4.7074

Riyantoko, P. A., Fahruddin, T. M., Hindrayani, K. M., & Safitri, E. M. (2020). Analisis Prediksi Harga Saham Sektor Perbankan Menggunakan Algoritma Long-Short Terms Memory (LSTM). *Seminar Nasional Informatika (SEMNASIF)*, *1*(1), 427–435. http://www.jurnal.upnyk.ac.id/index.php/semnasif/article/view/4135

Suradiradja, K. H. (2022). Algoritme Machine Learning Multi-Layer Perceptron dan Recurrent Neural Network untuk Prediksi Harga Cabai Merah Besar di Kota Tangerang. *Faktor Exacta*, *14*(4), 194. https://doi.org/10.30998/faktorexacta.v14i4.10376

Syaidah, K., Chrisnanto, Y. H., & Abdillah, G. (2020). Prediksi Harga Sembako di DKI Jakarta Menggunakan Artificial Neural Network. *JUMANJI (Jurnal Masyarakat Informatika Unjani)*, *3*(02), 136. https://doi.org/10.26874/jumanji.v3i02.63

Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for Time Series Prediction in Indian Stock Market. *Procedia Computer Science*, *167*, 2091–2100. https://doi.org/10.1016/j.procs.2020.03.257

Yanwardhana, E. (2022, June 15). *Ternyata Gegara Ini Harga Cabai Terbang Sampai Ratusan Ribu*. CNBC Indonesia. https://www.cnbcindonesia.com/news/20220615103728-4-347195/ternyata-gegara-ini-harga-cabai-terbang-sampai-ratusan-ribu

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 48 – 62
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Extreme Gradient Boosting Model with SMOTE for Heart Disease Classification

Ahmad Ubai Dullah [(1)*], Aditya Yoga Darmawan [(2)], Dwika Ananda Agustina Pertiwi [(3)], Jumanto Unjung [(4)]
[1,2,4] Department of Computer Science, Universitas Negeri Semarang, Semarang, Indonesia
[3] Faculty of Technology Management and Business, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia
e-mail : {ubaid,darmoenoyoga}@students.unnes.ac.id, hp220072@student.uthm.edu.my, jumanto@mail.unnes.ac.id.
* Corresponding author.
This article was submitted on 18 October 2024, revised on 9 November 2024, accepted on 10 November 2024, and published on 31 January 2025.

Abstract

Heart disease is one of the leading causes of death worldwide. According to data from the World Health Organisation (WHO), the number of victims who die from heart disease reaches 17.5 million people every year. However, the method of diagnosing heart disease in patients is still not optimal in determining the right treatment. Along with technology development, various models of machine learning algorithms and data processing techniques have been developed to find models that can produce the best precision in classifying heart disease. This research aims to develop a machine learning algorithm model in classifying heart disease to improve the effectiveness of diagnosis and help in determining the right treatment for patients. This research also aims to overcome the limitations of accuracy in existing diagnosis methods by identifying models capable of providing the best results in processing and analysing health data, especially in terms of heart disease classification. In this study, the XGBoost model was identified as the most superior, with an accuracy of 99%. These results show that the XGBoost model has a higher accuracy rate than previous methods, making it a promising solution to improve the accuracy of future heart disease diagnosis and classification.

Keywords: Heart Disease, SMOTE, XGBoost, KNN, SVM

Abstrak

Penyakit jantung adalah salah satu penyebab utama kematian di seluruh dunia. Menurut data dari World Health Organisation (WHO), jumlah korban yang meninggal akibat penyakit jantung mencapai 17,5 juta orang setiap tahunnya. Meski demikian, metode diagnosis penyakit jantung pada pasien masih belum optimal dalam menentukan penanganan yang tepat. Seiring dengan perkembangan teknologi, berbagai model algoritma *machine learning* dan teknik pengolahan data telah dikembangkan untuk menemukan model yang dapat menghasilkan akurasi terbaik dalam mengklasifikasikan penyakit jantung. Penelitian ini bertujuan untuk mengembangkan model algoritma *machine learning* dalam mengklasifikasikan penyakit jantung, sehingga dapat meningkatkan efektifitas diagnosa dan membantu dalam menentukan pengobatan yang tepat bagi pasien. Penelitian ini juga bertujuan untuk mengatasi keterbatasan akurasi pada metode diagnosis yang sudah ada, dengan cara mengidentifikasi model yang mampu memberikan hasil terbaik dalam mengolah dan menganalisa data kesehatan, khususnya dalam hal klasifikasi penyakit jantung. Pada penelitian ini, model XGBoost diidentifikasi sebagai model yang paling unggul, dengan akurasi sebesar 99%. Hasil ini menunjukkan bahwa model XGBoost memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan metode-metode sebelumnya, sehingga dapat menjadi solusi yang menjanjikan dalam meningkatkan akurasi diagnosis dan klasifikasi penyakit jantung di masa depan.

Kata Kunci: Penyakit Jantung, SMOTE, KNN, XGBoost, SVM

1. INTRODUCTION

Heart disease is one of the diseases that is considered to be the main cause of death of a person. Victims of heart disease and stroke are as many as 17.5 million people each year around the world, according to reports from the World Health Organization (WHO) (Baccouche et al., 2020; Xu et al., 2022). Heart disease is a collection of several conditions that affect human heart health (Benhar et al., 2020; Matin Malakouti, 2023). Some of these conditions include diseases of the blood vessels such as heart attack, stroke, heart failure, and arrhythmia. (El-Sofany, 2024; Radhika & Thomas George, 2021; Subathra & Sumathy, 2024). Two terms usually confuse most people, namely, the terms “heart disease” and ‘cardiovascular disease’, which is a situation that can cause heart attack, stroke, and chest pain (Maity et al., 2023; Pan et al., 2020). With the development of science, collecting data on heart disease is easier to obtain and analyse, helping to develop early diagnosis of heart disease (Ammar et al., 2021; Hossain et al., 2023).

In properly diagnosing patients with heart disease, it is necessary to classify heart disease. Research has been conducted on the classification of heart disease by J. P. Li et al. (2020) in 2020 using several machine learning classification models such as Naive Bayes, Support Vector Machine, Logistic Regression, Artificial Neural Network, Decision Tree, And k-Nearest Neighbor which are combined with several feature extractions to assist in data processing. Feature extraction is used to extract features from data that will be used to determine classification parameters. The results obtained have the best accuracy of 92.37% from the SVM model with FCMIM feature extraction. Another research by El-Sofany (2024) aims to employ three different feature selections such as chi-square, analysis of variance (ANOVA), and mutual information. This study also uses various machine learning such as Naive-Bayes, Support Vector Machine (SVM), Voting, XGBoost, AdaBoost, bagging, Decision Tree, K-Nearest Neighbor, Random Forest, Logistic Regression to classify heart disease. Using the SF-2 feature subset that contains 10 of 14 features and XGBoost with SMOTE to oversample the imbalanced data set from the combined Cleaveland Heart Disease Dataset and private dataset, the model reached an accuracy of 97,35%. Manikandan et al. (2024) Using Boruta feature selection and comparing 5 Machine Learning model performance such as Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, and XGBoost. The Cleaveland Heart Disease Data Set (Ashtaiwi et al., 2024) was used to train and test the model. The study achieved an accuracy of 88,52% using logistic regression. Another thing from this paper is that Boruta Feature Selection also improve model accuracy for the Support Vector Machine and Decision Tree, but this feature selection method also lower accuracy for the Random Forest (Gárate-Escamila et al., 2020) and XGBoost model, while Logistic Regression receives no improvement on accuracy. Using a newer dataset from Maghdid & Rashid (2022), research from Anshori & Haris (2022) uses logistic regression, support vector machine (SVM) and Linear Discriminant Analysis (LDA) to classify heart disease. The data is considered clean, and the researcher did not specify the training and test split amount. Cross-validation was used to evaluate each models, and Logistic Regression was the best model in their research, reaching 81,35% accuracy.

However, the model used is not optimal enough to classify heart disease, so there is a need to increase the resulting accuracy. Machine learning classification methods are increasingly developing, and new models are starting to emerge that can produce more optimal accuracy. Therefore, an analysis is needed to compare the machine learning models that have been developed to obtain more accurate results. Some models that will be used in this study include XGBoost (Chen et al., 2022; Mamun et al., 2022; Muslim et al., 2023), Support Vector Machine (M. Li et al., 2021; Wazrah & Alhumoud, 2021), Decision Tree (Haznedar & Simsek, 2022; Huang & Chen, 2022), Naive Bayes (Gibson et al., 2020), Logistic Regression (Bengesi et al., 2023), and K-nearest Neighbor (Islam et al., 2023).

2. METHODS

The methods used in this study are several machine learning classification models, namely XGBoost, SVM, Decision Tree, Logistic Regression, KNN, and Naive Bayes. Before being processed with the research model, the data will be processed at the preprocessing stage with

several stages such as cleaning and replacing values with numeric. Then sampling is carried out with SMOTE, the data will be trained with the model and produce an evaluation matrix. The methods in this study will be explained in the next section and shown in Figure 1.

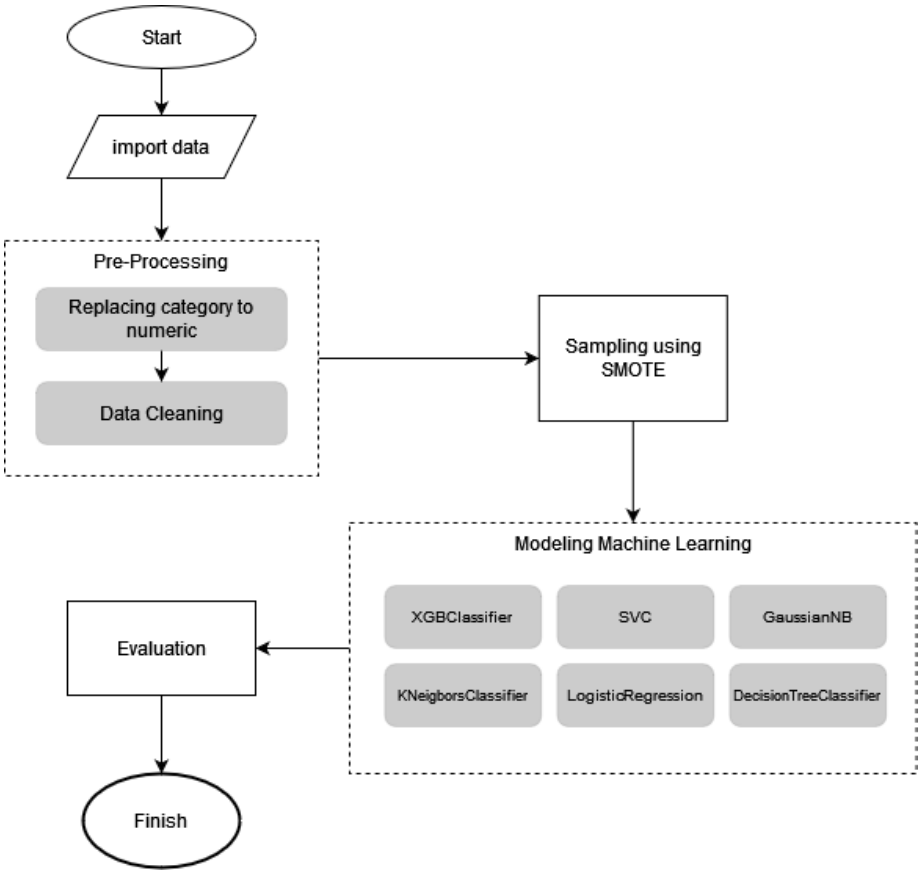


Figure 1 Flowchart for Proposed Method

2.1 Data Collection, Preprocessing and Sampling using SMOTE

The dataset used in this study is "An Extensive Dataset for the Heart Disease Classification System" released on Mendeley Data (Maghdid & Rashid, 2022). This dataset contains 1319 data with nine feature. There are 2 classes, 'positive' for CVD Positive with 810 data and 'negative' for CVD negative with 509 data. In data preprocessing, the data will be changed in value in the class column into binary form, which was originally positive and negative will change to 0 and 1. Negative will become 0, and positive will become 1. Because the data used in this study is text data, it is necessary to check for missing values and duplicates and remove them since missing and duplicate data will decrease model performance. Numerical Feature will be scaled using Formula 1.

$$z = \frac{(x - u)}{s} \quad (1)$$

With z is Scaled data, x is data before scaled, u is the mean of the data, and s is the standard deviation of the data. The Standard Scaler is performed using StandardScaler from the sklearn library.

After cleaning the data and checking the missing values, another Explorative Data Analysis is performed to check the balance of the data. Unbalanced data will affect the results obtained by the classification model. Methods to overcome data imbalance can use sampling. One library that

can be used is SMOTE (Sridhar & Sanagavarapu, 2021). SMOTE is a method for creating data samples to adjust the most data from each category to produce a good data balance. The data used in this study is unbalanced in the category for the class column. Based on the distribution of the class column, there are 61.4% of data with a value of 1 (Positive) and 38.6% of data with a value of 0 (negative). The distribution of data is uneven and needs to be balanced in order to get maximum results.

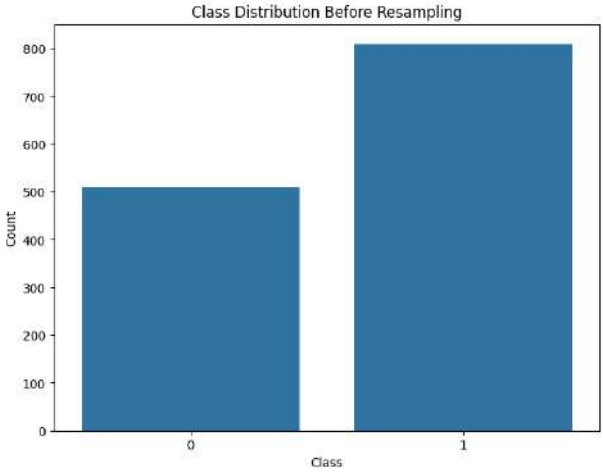


Figure 2 Before Resampling

2.2 Modeling Machine Learning

Modeling process begins with the process of importing machine learning classifier libraries, such as XGBClassifier for XGBoost, SVC for SVM, KNeigborClassifier for KNN, LogisticRegression for Logistic Regression, DecisionTreeClassifier for Decision Tree, and GaussianNB for Naive Bayes. Firstly, one of the machine learning algorithms, Naive Bayes, is based on the Bayes theory and assumes that every feature is independent (naive assumption) (El-Sofany, 2024). Even if the features are frequently not entirely independent, this algorithm is quite robust and effective, especially regarding text classification, such as spam filtering, sentiment analysis, and document generation. Gaussian Naive Bayes (GaussianNB) (Ningsih et al., 2024) is an initialization model. This algorithm summarizes that some parameters agree with a Gaussian (normal) distribution. Gaussian Naive Bayes is typically used when the fit is continuous.

Secondly, Support Vector Machine (SVM) is a machine learning algorithm that is highly effective for classification and regression tasks (Bengesi et al., 2023). SVM operates by searching for a hyperplane that maximizes the margin of error for each data set. This makes SVM extremely effective at solving classification problems, particularly when data cannot be processed linearly (Obiedat et al., 2022). Initialization of the SVM model using a linear kernel (kernel='linear') (Rofik et al., 2024). The type of hyperplane that is used to sift data is determined by the kernel. In this case, the linear kernel that is evaluated means that the model will search for linear terms or linear polynomials.

K-Nearest Neighbors (KNN) is a machine learning algorithm for classification and regression. KNN (El-Sofany, 2024) operates according to the following principle: when new data is provided, KNN determines the class or value of the data based on the k data matched in the training dataset. K-Nearest Neighbors (jabbar et al., 2013) model is analyzed with the parameter n_neighbors=5. Accordingly, the model will employ five lateral tangents to determine the new data set. KNN can't learn like other algorithms; instead, it just provides long-term data that can be used for prediction. The Logistic Regression algorithm is a machine learning technique used for classification (Patidar et al., 2022), primarily for binary classification problems (two classes). Despite being called "regression," logistic regression is a classification model rather than a linear regression. The

model is trained with a 'max_iter=1000' parameter. This parameter sets the maximum number of iterations for the optimization algorithm used in model training.

A decision Tree is a machine learning algorithm for regression and classification. Decision trees break down datasets into smaller subsets based on the current feature until they reach the end (leaf from tree) (Huang & Chen, 2022). This graph's structure is composed of single-simulation (nodes) that monitor a feature or attribute, branch-branch (branches) that monitor a feature's values, and leaf-branch (leaves) that monitor a class or prediction. Decision Tree Classifier from the scikit_learn library is used to create a probabilistic model for classification (Oh, 2021). Lastly, Extreme Gradient Boosting, or XGBoost, is a popular and effective machine learning algorithm for classification and regression tasks (El-Sofany, 2024). XGBoost is a single boosting technique that uses ensemble learning to maximize prediction model performance. XGBoost model initialization for classification using XGBClassifier from the XGBoost source (Zhang & Gong, 2020). This parameter uses the default value from the XGBoost.

2.3 Evaluation Model

The evaluation model uses a confusion matrix with the following composition, F1-score accuracy, recall, and precision. This performance analysis focuses on the accuracy produced by the proposed model compared to previous research. The mathematical formula 2, 3, 4, and 5 is used to analyse the results using confusion matrix.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100 \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F1\ Score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \quad (5)$$

True Positive (TP) refers to the number of correct instances identified as positive. True Negative (TN) represents the number of incorrect instances identified as negative. False Positive (FP) occurs when correct instances are mistakenly classified as positive, while False Negative (FN) happens when incorrect instances are misclassified as positive.

3. RESULTS AND DISCUSSION

3.1 Result

The results of this study contain the results of processing on the research model. Several stages of the process are passed, such as preprocessing and the results of model testing and evaluation.At the data cleaning stage, checking and cleaning the data is carried out so that the research model can process it. The results of the data-cleaning process can be seen in Table 1. It can be seen that the data is clean from missing values. However, the range age column is not used because it is better to use the age column. Therefore, the column will be dropped, and duplicate data will be cleaned up.

The data that is changed is the data in the class column used as a label. Here the class column contains data in the form of objects with contents, positive and negative. Then, the data must be converted to a numeric form to facilitate data processing in the research model. So, the data will be changed to 1 for positive and 0 for negative. Replacing value results can be seen in Table 2. The algorithm addresses the class imbalance in the target variable using the Synthetic Minority Over-sampling Technique (SMOTE). The training data is first divided into the target variable (y)

and characteristics (X). All columns are included in the features, except the target column "class," and the data from that column is contained in column y. To balance the class distribution, synthetic samples of the minority class are subsequently created using SMOTE. Results are guaranteed to be consistent when random_state=42 is used. The original dataset is smaller than the resampled data, X_resampled and y_resampled. Lastly, the code prints the shapes from the original and resampled datasets to show the modifications. The results of the sampling can be seen in Table 3.

Table 1 Result from Cleaning Data

Column Name	Value
age	0
gender	0
impluse	0
pressurehight	0
pressurelow	0
glucose	0
kcm	0
troponin	0
class	0
Age_Range	0

Table 2 Result from Replacing Value

Class before replacing	Class after replacing
Negative	0
Positive	1

Table 3 Result from Resampling

Before sampling using SMOTE		After sampling using SMOTE	
Class	**Count**	**Class**	**Count**
1	647	1	647
0	408	0	647

Data sampled with SMOTE will follow the largest amount of data, which is 647 data at value 1. Therefore, the value 0 data will change to 647, originally 408 data, that way the data used will be balanced. As for result without SMOTE, the training data that is not sampled using SMOTE oversampling has been tested using the research model with the results as a classification report as follows.

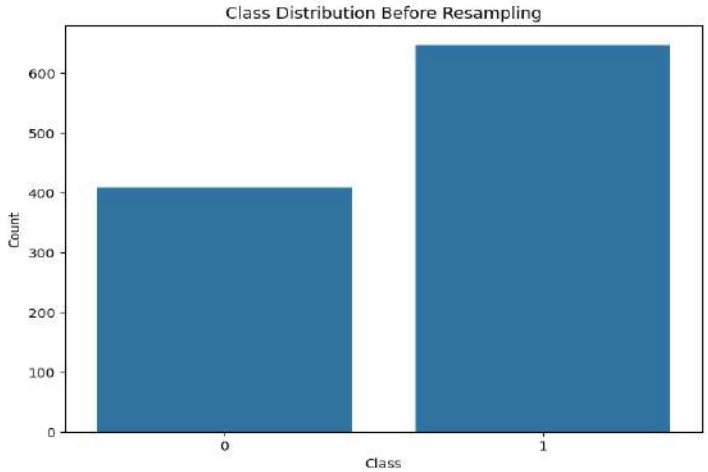


Figure 3 Before Resampling

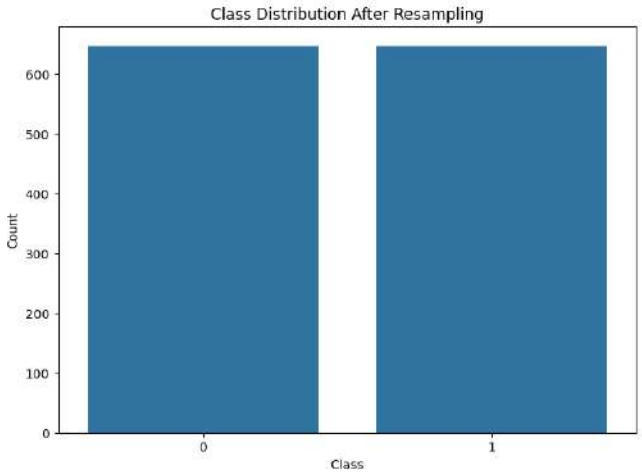


Figure 4 After Resampling

3.1.1 Naïve Bayes without SMOTE

The results obtained by the Naïve Bayes research model using data that is not oversampled can be seen in Table 4 as a classification report. The classification report shows the model's performance in distinguishing between two classes, '0' and '1'. For class 0, the model obtained an average precision (64.52%) but a very high recall rate (99.01%), which means that the model identified the largest number of correct examples in this class, resulting in a decent F1 value of 0.7812. For class 1, the model achieved high precision (99.08%) but lower recall (66.26%), meaning that the model missed a few correct examples from this class, with an F1 value of 0.7941. Overall, the model had a precision of 78.79%. The macro averages (precision 0.8180, recall 0.8263, F1 score 0.7877) show a balanced performance across the two classes, while the weighted average considers the class distribution.

Table 4 Result Naive Bayes without SMOTE

	Precision	Recall	F1-score	Support
0	0.6452	0.9901	0.7812	101
1	0.9908	0.6626	0.7941	163
Accuracy			0.7879	264
Macro avg	0.8180	0.8263	0.7877	264
Weighted avg	0.8586	0.7879	0.7892	264

3.1.2 Support Vector Machine without SMOTE

Table 5 Result from SVM without SMOTE

	Precision	Recall	F1-score	Support
0	0.7282	0.7426	0.7353	101
1	0.8385	0.8282	0.8333	163
Accuracy			0.7955	264
Macro avg	0.7833	0.7854	0.7843	264
Weighted avg	0.7963	0.7955	0.7958	264

The results obtained by the support vector machine research model using data that is not oversampled can be seen in Table 5 as a classification report. This classification report shows the performance of the model for two classes, '0' and '1'. For class 0, the model has an accuracy of 72.82% and a recall of 74.26%, resulting in an F1 score of 0.7353. For class 1, the model achieved a higher accuracy of 83.85% and a higher recall of 82.82% with an F1 value of 0.8333. The overall accuracy of this model was 79.55%. The macro-mean (precision 0.7833, recall

0.7854, F1-score 0.7843) showed a balanced performance for both classes, while the weighted mean (precision 0.7963, recall 0.7955, F1-score 0.7958) reflected the class distribution, suggesting that the model performed quite well for both classes, with a slight advantage for the prediction of class 1.

3.1.3 K-Nearest Neighbor without SMOTE

The results obtained by the k-nearest neighbor research model using data that is not oversampled can be seen in Table 6 as a classification report. This classification report shows the model's performance for classes '0' and '1'. For class 0, the model has an accuracy of 57.45% and a recognition rate of 53.47%, giving an F1 value of 0.5538. Class 1 has a higher accuracy of 72.35% and a recognition rate of 75.46%, giving an F1 value of 0.7387. The overall accuracy was 67.05%. The macro average (precision 0.6490, recall 0.6446, F1 score 0.6463) shows moderate performance for all classes, while the weighted average (precision 0.6665, recall 0.6705, F1 score 0.6680) reflects slightly better performance for class 1 due to greater support, indicating that the model generally prefers class 1 over class 0 in its predictions.

Table 6 Result from KNN without SMOTE

	Precision	Recall	F1-score	Support
0	0.5745	0.5347	0.5538	101
1	0.7235	0.7546	0.7387	163
Accuracy			0.6705	264
Macro avg	0.6490	0.6446	0.6463	264
Weighted avg	0.6665	0.6705	0.6680	264

3.1.4 Logistic Regression without SMOTE

The results obtained by the logistic regression research model using data that is not oversampled can be seen in Table 7 as a classification report. This classification report with precision, recall, and F1-score metrics for two classes labeled "0" and "1." For class "0," the precision is 0.7609, recall is 0.6931, and F1-score is 0.7254, based on 101 instances. For class "1," the precision is higher at 0.8198, with a recall of 0.8650 and an F1-score of 0.8318, based on 163 instances. The model's overall accuracy is 0.7992, indicating that it correctly classified approximately 79.92% of the samples. Additionally, the macro average (averaging both classes without considering class imbalance) and weighted average (considering class imbalance) for F1-scores are 0.7836 and 0.7973, respectively, reflecting consistent performance across both classes.

Table 7 Result Logistic Regression Without SMOTE

	Precision	Recall	F1-score	Support
0	0.7609	0.6931	0.7254	101
1	0.8198	0.8650	0.8318	163
Accuracy			0.7992	264
Macro avg	0.7903	0.7790	0.7836	264
Weighted avg	0.7972	0.7992	0.7973	264

3.1.5 Decision Tree without SMOTE

The results obtained by the decision tree research model using data that is not oversampled can be seen in Table 8 as a classification report. This classification report shows that the model decision tree has high precision, recall, and F1 scores for both classes. For class "0," precision, recall, and F1-score are all 0.9703, based on 101 instances. For class "1," these metrics are slightly higher, with precision, recall, and F1-score of 0.9816, based on 163 instances. The model achieves an overall accuracy of 0.9773, indicating that it correctly classified approximately 97.73% of the samples. The macro average and weighted average F1-scores are around 0.9759 and 0.9773, respectively, reflecting consistently high performance across both classes.

Table 8 Result Decision Tree without SMOTE

	Precision	Recall	F1-score	Support
0	0.9703	0.9703	0.9703	101
1	0.9816	0.9816	0.9816	163
Accuracy			0.9773	264
Macro avg	0.9759	0.9759	0.9759	264
Weighted avg	0.9773	0.9773	0.9773	264

3.1.6 Extreme Gradient Boosting without SMOTE

The results obtained by the extreme gradient boosting research model using data that is not oversampled can be seen in Table 9 in the form of a classification report. This classification report demonstrates excellent performance across both classes. For class "0," the precision is 0.9800, recall is 0.9703, and the F1-score is 0.9751, based on 101 instances. For class "1," the precision is slightly higher at 0.9817, with a recall of 0.9877 and an F1-score of 0.9847, based on 163 instances. The model achieves an overall accuracy of 0.9811, indicating it correctly classified approximately 98.11% of the samples. The macro average and weighted average F1-scores are 0.9799 and 0.9810, respectively, reflecting strong and consistent performance across both classes.

Table 9 Result XGBoost without SMOTE

	Precision	Recall	F1-score	Support
0	0.9800	0.9703	0.9751	101
1	0.9817	0.9877	0.9847	163
Accuracy			0.9811	264
Macro avg	0.9809	0.9790	0.9799	264
Weighted avg	0.9811	0.9811	0.9810	264

As a result with SMOTE, the model is also trained with training data oversampled using SMOTE, which produces the following classification report. The results of this case are depicted as follows.

3.1.7 Naïve Bayes with SMOTE

The results of testing the naive bayes model with data balanced with SMOTE obtained classification report results, which can be seen in Table 10. The model achieved an accuracy of 80%, indicating that 80% of the predictions were correct. For class 0, the precision is 0.65, meaning 65% of the predicted class 0 instances were correct, with a high recall of 0.99, showing that almost all actual class 0 instances were correctly identified. For class 1, the precision is 0.99, indicating strong performance in predicting class 1, but the recall is lower at 0.68, meaning that only 68% of actual class 1 instances were correctly predicted. The F1 scores, which balance precision and recall, are 0.79 for class 0 and 0.80 for class 1. The macro average, which averages precision, recall, and F1-score across both classes without accounting for class imbalance, shows precision at 0.82, recall at 0.83, and F1-score at 0.80. The weighted average, which considers class imbalance, gives similar values with precision at 0.86, recall at 0.80, and F1-score at 0.80. The model performs well for class 0 but shows weaker recall for class 1, indicating room for improvement in predicting that class.

Table 10 Result from Naive Bayes with SMOTE

	Precision	Recall	F1-score	Support
0	0.6536	0.9901	0.7874	101
1	0.9910	0.6748	0.8029	163
Accuracy			0.7955	264
Macro avg	0.8223	0.8325	0.7952	264
Weighted avg	0.8619	0.7955	0.7970	264

3.1.8 Support Vector Machine with SMOTE

The results of testing the support vector machine model with data balanced with SMOTE obtained classification report results, which can be seen in Table 11. The performance metrics of an SVM (Support Vector Machine) model achieved an accuracy of 78%. This indicates that the model correctly classified 78% of the instances. For class 0, the model has a precision of 0.62, meaning 62% of the predicted class 0 instances were correct, and a high recall of 0.97, indicating that 97% of the actual class 0 instances were accurately identified. For class 1, the precision is higher at 0.97, but the recall is lower at 0.66, meaning only 66% of the actual class 1 instances were correctly predicted. The F1 scores, which balance precision and recall, are 0.77 for class 0 and 0.79 for class 1. The macro average for precision, recall, and F1-score across both classes is 0.81, 0.82, and 0.78, respectively. The weighted average, which accounts for class imbalance, has slightly different results, where the precision, recall, and F1-score are 0.85, 0.78, 0.78.

Table 11 Result from SVM with SMOTE

	Precision	Recall	F1-score	Support
0	0.6405	0.9703	0.7717	101
1	0.9730	0.6626	0.7883	163
Accuracy			0.7803	264
Macro avg	0.8067	0.8164	0.7800	264
Weighted avg	0.8454	0.7803	0.7819	264

3.1.9 K-Nearest Neighbor with SMOTE

The results of testing the K-nearest Neighbor model with data balanced with SMOTE obtained classification report results, which can be seen in Table 12. The performance metrics of a K-Nearest Neighbors (KNN) classification model achieved an accuracy of approximately 0.65 (65%). This means the model correctly predicted the class for 68% of the instances. For class 0, the precision is 0.54, indicating that 54% of the predicted class 0 instances were correct, while the recall is 0.65, meaning 65% of the actual class 0 instances were identified correctly. For class 1, the precision is 0.75, but the recall is lower at 0.65, meaning only 65% of the actual class 1 instances were predicted correctly. The F1 scores, which balance precision and recall, are 0.59 for class 0 and 0.70 for class 1. The macro average for precision, recall, and F1-score across both classes is 0.64, 0.65, and 0.64, indicating that the model performs similarly for both classes. The weighted averages account for class imbalance and result in 0.67 for precision, 0.65 in recall, and 0.64 in F1-score. Overall, the KNN model has moderate performance, showing some difficulty distinguishing between the two classes, particularly with a lower recall for class 1. This indicates room for improvement in the model's predictive capability.

Table 12 Result from KNN with SMOTE

	Precision	Recall	F1-score	Support
0	0.5366	0.6535	0.5893	101
1	0.7518	0.6503	0.6974	163
Accuracy			0.6515	264
Macro avg	0.6442	0.6519	0.6433	264
Weighted avg	0.6694	0.6515	0.6560	264

3.1.10 Logistic Regression with SMOTE

The results of testing the logistic regression model with data balanced with SMOTE obtained classification report results, which can be seen in Table 13. The performance metrics of the Logistic Regression classification model achieved an accuracy of approximately 0.79 (79%). This means the model correctly predicted the class for 79% of the instances. For class 0, the precision is 0.67, indicating that 67% of the predicted class 0 instances were correct, while the recall is

0.86, meaning 86% of the actual class 0 instances were identified correctly. For class 1, the precision is 0.90, but the recall is lower at 0.74, meaning only 74% of the actual class 1 instances were predicted correctly. The F1 scores, which balance precision and recall, are 0.76 for class 0 and 0.81 for class 1. The macro average for precision, recall, and F1-score across both classes is 0.79, 0.80, and 0.78, indicating that the model performs similarly for both classes. The weighted averages, which account for class imbalance, also result in 0.81 for precision, 0.79 in recall, and 0.79 in F1-score. Overall, the Logistic Regression model has moderate performance, showing difficulty distinguishing between the two classes, particularly with a lower precision for class 0. This indicates room for improvement in the model's predictive capability.

Table 13 Result from Logistic Regression with SMOTE

	Precision	Recall	F1-score	Support
0	0.6744	0.8614	0.7565	101
1	0.8963	0.7423	0.8121	163
Accuracy			0.7879	264
Macro avg	0.7854	0.8019	0.7843	264
Weighted avg	0.8114	0.7879	0.7903	264

3.1.11 Decision Tree with SMOTE

The results of testing the decision tree model with data balanced with SMOTE obtained classification report results, which can be seen in Table 14. Performance summary for a decision tree model. This model achieved a high accuracy of approximately 0.981. The table includes the performance metrics such as 'precision', 'recall', 'f1-score', and 'support' for two classes, labeled '0' and '1'. For Class 0, the precision is 0.98, recall is 0.97, and the f1-score is 0.98. Class 1 shows a precision of 0.98, a recall of 0.99, and an f1-score of 0.99. The model performs exceptionally with high-performance metrics for both macro and weighted average calculation.

Table 14 Result Decision Tree with SMOTE

	Precision	Recall	F1-score	Support
0	0.9800	0.9703	0.9751	101
1	0.9817	0.9877	0.9847	163
Accuracy			0.9811	264
Macro avg	0.9809	0.9790	0.9799	264
Weighted avg	0.9811	0.9811	0.9810	264

3.1.12 Extreme Gradient Boosting with SMOTE

The results of testing the extreme gradient boosting model with data balanced with SMOTE obtained classification report results, which can be seen in Table 15. Performance metrics for an XGBoost model, which has achieved an impressive accuracy of approximately 0.985. The metrics detailed include 'precision', 'recall', and 'f1-score' for two classes, labeled as '0' and '1'. Both classes show outstanding performance with a precision, recall, and f1-score of around 0.98 - 0.99. This summary indicates a highly effective model performance across all evaluated categories.

Table 15 Result XGBoost With SMOTE

	Precision	Recall	F1-score	Support
0	0.9848	0.9802	0.9802	101
1	0.9877	0.9877	0.9877	163
Accuracy			0.9848	264
Macro avg	0.9840	0.9840	0.9840	264
Weighted avg	0.9848	0.9848	0.9848	264

3.2 Discussion

The discussion will be a comparison between research models that have been trained using data that has not been balanced with SMOTE and after being balanced with SMOTE. Then, the best model is used as a proposed model. The proposed model will be compared with the previous research model. A comparison of research models can be seen in Table 16. Based on the comparison table of each research model, the XGBoost with the SMOTE model has very good results. The accuracy obtained reaches 98.48% with Precision, Recall, and F1-Score also around 98%. It can be ascertained that the XGBoost research model is better than other research models. So, it can be said that the XGBoost with SMOTE model is the proposed model. Another thing to point out is SMOTE can increase performance of model, with increased performance in models like Naïve Bayes, Decision Tree, and XGBoost. However, model KNN, Logistic Regression and SVM saw no increase in performance.

Table 16 Comparison Result

	Without SMOTE				With SMOTE			
Proposed Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Naïve Bayes	78.79	81.80	82.63	78.77	79.55	82.23	81.64	78.00
KNN	67.05	64.90	64.46	64.63	65.15	64.42	65.19	64.33
Logistic Regression	79.92	79.03	77.90	78.36	78.79	78.54	80.19	78.43
SVM	79.55	78.33	78.54	78.43	78.03	80.67	81.64	78.00
Decision Tree	97.73	97.59	97.59	97.59	98.11	98.09	97.90	97.99
XGBoost	98.11	98.09	97.90	97.99	98.48	98.40	98.40	98.40

Then, the proposed model will be compared with models from previous research. The comparison table of the proposed model with previous research models can be seen in Table 17. The SMOTE technique on XGBoost improves the performance compared to XGBoost itself and surpasses the more complex or simpler methods used by other researchers in the table. This confirms the importance of a good approach in preparing data and choosing the right algorithm for a particular type of data.

Table 17 Comparison with Previous Research

Author	Model Algorithm	Result
El-Sofany (2024)	XGBoost with sampling SMOTE	97.57%
J. P. Li et al. (2020)	FCMIM-SVM	92.37%
Anshori & Haris (2022)	Logistic Regression	81.3%
Proposed Method	XGBoost+SMOTE	98.48%

4. CONCLUSIONS

The results of this study demonstrate that the proposed classification model for heart disease, which integrates the Extreme Gradient Boosting (XGBoost) algorithm with Synthetic Minority Over-sampling Technique (SMOTE), yields superior performance compared to other machine learning models tested. The model achieved a classification accuracy of 98.48%, with precision, recall, and F1-score values consistently above 98%, indicating a high level of reliability and generalizability. These results substantiate the effectiveness of combining advanced ensemble learning with appropriate resampling techniques in addressing class imbalance issues within medical datasets.

Furthermore, the comparative analysis reveals that the XGBoost-SMOTE model outperforms several other baseline classifiers, including Support Vector Machine, Naive Bayes, Logistic Regression, K-Nearest Neighbors, and Decision Tree, both in pre- and post-resampling conditions. The findings also highlight that while SMOTE positively impacts model performance

across most algorithms, its integration with XGBoost delivers the most substantial improvement, thus reinforcing its suitability for the classification of complex, imbalanced clinical data.

When compared to models from prior research, the proposed model exhibits an enhancement in classification performance, surpassing the highest previously reported accuracy of 97.57%. This underscores the significance of meticulous model selection and data preprocessing strategies in developing predictive tools for clinical decision support. Given its empirical robustness and superior accuracy, the XGBoost-SMOTE model proposed in this study holds considerable potential for adoption in real-world diagnostic systems to support early and accurate detection of heart disease.

REFERENCES

Ammar, A., Bouattane, O., & Youssfi, M. (2021). Automatic Cardiac Cine MRI Segmentation and Heart Disease Classification. *Computerized Medical Imaging and Graphics*, *88*(2020), 101864. https://doi.org/10.1016/j.compmedimag.2021.101864

Anshori, M., & Haris, M. S. (2022). Predicting Heart Disease Using Logistic Regression. *Knowledge Engineering and Data Science*, *5*(2), 188. https://doi.org/10.17977/um018v5i22022p188-196

Ashtaiwi, A., Khalifa, T., & Alirr, O. (2024). Enhancing Heart Disease Diagnosis Through ECG Image Vectorization-Based Classification. *Heliyon*, *10*(18), e37574. https://doi.org/10.1016/j.heliyon.2024.e37574

Baccouche, A., Garcia-Zapirain, B., Castillo Olea, C., & Elmaghraby, A. (2020). Ensemble Deep Learning Models for Heart Disease Classification: A Case Study from Mexico. *Information*, *11*(4), 207. https://doi.org/10.3390/info11040207

Bengesi, S., Oladunni, T., Olusegun, R., & Audu, H. (2023). A Machine Learning-Sentiment Analysis on Monkeypox Outbreak: An Extensive Dataset to Show the Polarity of Public Opinion from Twitter Tweets. *IEEE Access*, *11*, 11811–11826. https://doi.org/10.1109/ACCESS.2023.3242290

Benhar, H., Idri, A., & Fernández-Alemán, J. L. (2020). Data Preprocessing for Heart Disease Classification: A Systematic Literature Review. *Computer Methods and Programs in Biomedicine*, *195*, 105635. https://doi.org/10.1016/j.cmpb.2020.105635

Chen, L., Ji, P., & Ma, Y. (2022). Machine Learning Model for Hepatitis C Diagnosis Customized to Each Patient. *IEEE Access*, *10*(10), 106655–106672. https://doi.org/10.1109/ACCESS.2022.3210347

El-Sofany, H. F. (2024). Predicting Heart Diseases Using Machine Learning and Different Data Classification Techniques. *IEEE Access*, *12*(10), 106146–106160. https://doi.org/10.1109/ACCESS.2024.3437181

Gárate-Escamila, A. K., Hassani, A. H. El, & Andrès, E. (2020). Classification Models for Heart Disease Prediction Using Feature Selection and PCA. *Informatics in Medicine Unlocked*, *19*, 100330. https://doi.org/10.1016/j.imu.2020.100330

Gibson, S., Issac, B., Zhang, L., & Jacob, S. M. (2020). Detecting Spam Email with Machine Learning Optimized with Bio-Inspired Metaheuristic Algorithms. *IEEE Access*, *8*, 187914–187932. https://doi.org/10.1109/ACCESS.2020.3030751

Haznedar, B., & Simsek, N. Y. (2022). A Comparative Study on Classification Methods for Renal Cell and Lung Cancers Using RNA-Seq Data. *IEEE Access*, *10*(10), 105412–105420. https://doi.org/10.1109/ACCESS.2022.3211505

Hossain, Md. I., Maruf, M. H., Khan, Md. A. R., Prity, F. S., Fatema, S., Ejaz, Md. S., & Khan, Md. A. S. (2023). Heart Disease Prediction Using Distinct Artificial Intelligence Techniques: Performance Analysis and Comparison. *Iran Journal of Computer Science*, *6*(4), 397–417. https://doi.org/10.1007/s42044-023-00148-7

Huang, Z., & Chen, D. (2022). A Breast Cancer Diagnosis Method Based on VIM Feature Selection and Hierarchical Clustering Random Forest Algorithm. *IEEE Access*, *10*, 3284–3293. https://doi.org/10.1109/ACCESS.2021.3139595

Islam, N., Fatema-Tuj-Jahra, M., Hasan, Md. T., & Farid, D. Md. (2023). KNNTree: A New Method to Ameliorate K-Nearest Neighbour Classification Using Decision Tree. *2023 International*

Conference on Electrical, Computer and Communication Engineering (ECCE), 1–6. https://doi.org/10.1109/ECCE57851.2023.10101569

jabbar, M. A., Deekshatulu, B. L., & Chandra, P. (2013). Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm. *Procedia Technology*, *10*, 85–94. https://doi.org/10.1016/j.protcy.2013.12.340

Li, J. P., Haq, A. U., Din, S. U., Khan, J., Khan, A., & Saboor, A. (2020). Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare. *IEEE Access*, *8*(2), 107562–107582. https://doi.org/10.1109/ACCESS.2020.3001149

Li, M., Ma, X., Chen, C., Yuan, Y., Zhang, S., Yan, Z., Chen, C., Chen, F., Bai, Y., Zhou, P., Lv, X., & Ma, M. (2021). Research on the Auxiliary Classification and Diagnosis of Lung Cancer Subtypes Based on Histopathological Images. *IEEE Access*, *9*, 53687–53707. https://doi.org/10.1109/ACCESS.2021.3071057

Maity, A., Pathak, A., & Saha, G. (2023). Transfer Learning Based Heart Valve Disease Classification from Phonocardiogram Signal. *Biomedical Signal Processing and Control*, *85*(2022), 104805. https://doi.org/10.1016/j.bspc.2023.104805

Mamun, M., Farjana, A., Al Mamun, M., & Ahammed, M. S. (2022). Lung Cancer Prediction Model Using Ensemble Learning Techniques and a Systematic Review Analysis. *2022 IEEE World AI IoT Congress (AIIoT)*, *2022*, 187–193. https://doi.org/10.1109/AIIoT54504.2022.9817326

Manikandan, G., Pragadeesh, B., Manojkumar, V., Karthikeyan, A. L., Manikandan, R., & Gandomi, A. H. (2024). Classification Models Combined with Boruta Feature Selection for Heart Disease Prediction. *Informatics in Medicine Unlocked*, *44*(2023), 101442. https://doi.org/10.1016/j.imu.2023.101442

Matin Malakouti, S. (2023). Heart Disease Classification Based on ECG Using Machine Learning Models. *Biomedical Signal Processing and Control*, *84*(2022), 104796. https://doi.org/10.1016/j.bspc.2023.104796

Muslim, M. A., Nikmah, T. L., Pertiwi, D. A. A., Subhan, Jumanto, Dasril, Y., & Iswanto. (2023). New Model Combination Meta-Learner to Improve Accuracy Prediction P2P Lending with Stacking Ensemble Learning. *Intelligent Systems with Applications*, *18*(2022), 200204. https://doi.org/10.1016/j.iswa.2023.200204

Ningsih, M. R., Unjung, J., Farih, H. al, & Muslim, M. A. (2024). Classification Email Spam Using Naive Bayes Algorithm and Chi-Squared Feature Selection. *Journal of Applied Intelligent System*, *9*(1), 74–87. https://doi.org/10.33633/JAIS.V9I1.9695

Obiedat, R., Qaddoura, R., Al-Zoubi, A. M., Al-Qaisi, L., Harfoushi, O., Alrefai, M., & Faris, H. (2022). Sentiment Analysis of Customers' Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution. *IEEE Access*, *10*, 22260–22273. https://doi.org/10.1109/ACCESS.2022.3149482

Oh, H. (2021). A YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model. *IEEE Access*, *9*, 144121–144128. https://doi.org/10.1109/ACCESS.2021.3121508

Pan, Y., Fu, M., Cheng, B., Tao, X., & Guo, J. (2020). Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on the Internet of Medical Things Platform. *IEEE Access*, *8*, 189503–189512. https://doi.org/10.1109/ACCESS.2020.3026214

Patidar, S., Kumar, D., & Rukwal, D. (2022). Comparative Analysis of Machine Learning Algorithms for Heart Disease Prediction. In *Advanced Production and Industrial Engineering* (pp. 64–69). https://doi.org/10.3233/ATDE220723

Radhika, R., & Thomas George, S. (2021). Heart Disease Classification Using Machine Learning Techniques. *Journal of Physics: Conference Series*, *1937*(1), 012047. https://doi.org/10.1088/1742-6596/1937/1/012047

Rofik, R., Hakim, R. A., Unjung, J., Prasetiyo, B., & Muslim, M. A. (2024). Optimization of SVM and Gradient Boosting Models Using GridSearchCV in Detecting Fake Job Postings. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, *23*(2), 419–430. https://doi.org/10.30812/matrik.v23i2.3566

Maghdid, S. S., & Rashid, T. A. (2022). *An Extensive Dataset for the Heart Disease Classification System*. *2*. https://doi.org/10.17632/65GXGY2NMG.2

Sridhar, S., & Sanagavarapu, S. (2021). Handling Data Imbalance in Predictive Maintenance for Machines Using SMOTE-Based Oversampling. *2021 13th International Conference on Computational Intelligence and Communication Networks (CICN)*, 44–49. https://doi.org/10.1109/CICN51697.2021.9574668

Subathra, R., & Sumathy, V. (2024). An Offbeat Bolstered Swarm Integrated Ensemble Learning (BSEL) Model for Heart Disease Diagnosis and Classification. *Applied Soft Computing*, *154*(2023), 111273. https://doi.org/10.1016/j.asoc.2024.111273

Wazrah, A. Al, & Alhumoud, S. (2021). Sentiment Analysis Using Stacked Gated Recurrent Unit for Arabic Tweets. *IEEE Access*, *9*, 137176–137187. https://doi.org/10.1109/ACCESS.2021.3114313

Xu, W., Yu, K., Ye, J., Li, H., Chen, J., Yin, F., Xu, J., Zhu, J., Li, D., & Shu, Q. (2022). Automatic Pediatric Congenital Heart Disease Classification Based on Heart Sound Signal. *Artificial Intelligence in Medicine*, *126*(2021), 102257. https://doi.org/10.1016/j.artmed.2022.102257

Zhang, D., & Gong, Y. (2020). The Comparison of LightGBM and XGBoost Coupling Factor Analysis and Prediagnosis of Acute Liver Failure. *IEEE Access*, *8*, 220990–221003. https://doi.org/10.1109/ACCESS.2020.3042848

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 63 – 73
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Class Weighting Approach for Handling Imbalanced Data on Forest Fire Classification Using EfficientNet-B1

Arvinanto Bahtiar [(1)*], Muhammad Ihsan Prawira Hutomo [(2)], Agung Widiyanto [(3)], Siti Khomsah [(4)]
Department of Data Science, Universitas Telkom Purwokerto, Purwokerto, Indonesia
e-mail : {arvinanto,muhammadihsanprawira,agungwdyy}@student.telkomuniversity.ac.id, sitijk@telkomuniversity.ac.id.
* Corresponding author.
This article was submitted on 20 October 2024, revised on 6 November 2024, accepted on 7 November 2024, and published on 31 January 2025.

Abstract

Wildfires threaten ecosystems and human safety, necessitating effective monitoring techniques. Detecting forest fires based on images of forest conditions could be a breakthrough. But, the model built from imbalanced data leads to low accuracy. This research addresses the challenge of class imbalance in multiclass classification for forest fire detection using the EfficientNet-B1 model. This research explores the implementation of class weighting to enhance model performance, particularly focusing on minority classes, namely: Fire and Smoke. A dataset of 7,331 training images was categorized into four classes. The results showed that employing the class weighting method achieved an accuracy of 90%. The training duration of 14 minutes and 45 seconds outperforms the data augmentation method in terms of time efficiency. This study contributes to the development of more effective methods for forest fire monitoring and provides insights for future research in machine learning applications in environmental contexts.

Keywords: Image Classification, Imbalanced Data, Efficientnet-B1, Forest Fire Detection

Abstrak

Kebakaran hutan menimbulkan ancaman besar terhadap ekosistem dan keselamatan manusia sehingga memerlukan teknik pemantauan yang efektif. Mendeteksi kebakaran hutan berdasarkan gambaran kondisi hutan bisa menjadi sebuah terobosan. Namun, model yang dibangun dari data yang tidak seimbang menyebabkan akurasi yang rendah. Penelitian ini bertujuan mengatasi ketidakseimbangan kelas dalam klasifikasi citra multi-kelas untuk deteksi kebakaran hutan menggunakan model EfficientNet-B1. Penerapan metode *Class Weighting* bertujuan meningkatkan kinerja model pada data tidak seimbang ini, terutama berfokus pada kelas minoritas yaitu kelas “Api” dan kelas “Asap”. Eksperimen ini menggunakan dataset terdiri 7,331 gambar untuk data pelatihan, yang dikategorikan ke dalam empat kelas. Hasil penelitian menunjukkan bahwa metode *Class Weighting* mencapai akurasi 90%. Sedangkan durasi pelatihan hanya memerlukan waktu 14 menit dan 45 detik, ini mengungguli metode augmentasi data dalam hal efisiensi waktu. Hasil penelitian ini berkontribusi pada pengembangan metode yang lebih efektif untuk pemantauan kebakaran hutan dan memberikan wawasan untuk penelitian di masa depan dalam aplikasi pembelajaran mesin dalam konteks lingkungan.

Kata Kunci: Klasifikasi Gambar, Data Tidak Seimbang, Efficientnet-B1, Kebakaran Hutan

1. INTRODUCTION

In recent years, climate change and human-induced factors have significantly impacted the environment. These events include heatwaves, droughts, dust storms, floods, hurricanes, and wildfires (Barmpoutis et al., 2020). Wildfires onseverely affect local and global ecosystems, resulting in significant damage to infrastructure, injuries, and loss of human life. Wildfires can be sparked by various human activities, including campfires, burning debris, unattended flames, smoking, the careless disposal of lit cigarettes, and natural causes like lightning (Chaturvedi et al., 2022). As a result, detecting fires and precisely monitoring type, size, and impact of disturbances across large areas are becoming increasingly crucial (Tanase et al., 2018). Today, the technology for detecting forest fires has advanced significantly through satellites, drones, and

advanced sensors. The use of deep learning methods is one such advancement towards detecting forest fires (Madhuri et al., 2024). Image-based monitoring technology offers a promising solution for automating the identification of fires and smoke through digital image processing. Multiclass image classification is essential for distinguishing between various objects and conditions related to wildfires. Therefore, developing accurate and efficient classification models can significantly contribute to disaster mitigation efforts.

Deep neural network models, such as EfficientNet-B1, have demonstrated outstanding performance in various image classification tasks (Frederich et al., 2024; Islam et al., 2024). EfficientNet is an innovation in network architecture that optimizes the scale of the model to achieve a balance between accuracy and computational efficiency (Raza et al., 2023). By utilizing intelligent scaling techniques, EfficientNet-B1 can better identify patterns in imagery compared to another model architecture, as it employs Compound Scaling, which scales all three dimensions depth, width, and image resolution simultaneously while maintaining this balance across the network (Papoutsis et al., 2023). This is especially important in forest fire monitoring, where the speed and accuracy of detection can be the difference between successful prevention or widespread damage. However, despite the great potential of these models, the challenges faced in multiclass classification often relate to class imbalance in the training data. Therefore, approaches are needed to ensure that all classes, including underrepresented classes, are well-learned by the model (Dogra et al., 2022; Rodríguez et al., 2020; Tanveer et al., 2021).

One approach that can address the problem of class imbalance is the application of class weights (Benkendorf et al., 2023). By assigning higher weights to underrepresented classes, the model can be trained to pay more attention to patterns within those classes. By assigning higher weights to underrepresented classes, the model can be trained to pay more attention to patterns within those classes. This not only improves accuracy for the minority class but also helps prevent the model from being biased towards the majority class (De Angeli et al., 2022). Many studies have demonstrated that class weights can enhance model performance in multiclass classification, particularly when the dataset is imbalanced (Fan et al., 2022; Zhao et al., 2020). In the context of forest fires, where images from the fire class may be much fewer than other classes, the application of class weights becomes especially relevant. Thus, this study aims to explore the implementation of class weights in multiclass image classification for forest fires and smoke.

In this study, we will analyze the effect of applying class weights on the performance of the EfficientNet-B1 model in detecting and classifying forest fire and smoke images. We will also compare the results obtained from the model using class weights with the model that uses data augmentation to address the imbalanced data issue. The evaluation method will include various metrics, such as accuracy, precision, recall, and F1-score, to provide a comprehensive picture of the model's performance. In addition, experiments will be conducted on a dataset of forest fire and smoke images collected from various sources to ensure diversity and representativeness. Through this analysis, we hope to provide deeper insights into how class weights can improve classification performance in this context. The results of this study are expected to contribute to developing a more effective fire monitoring system.

The results of this study will provide a stronger foundation for developing more accurate and efficient forest fire monitoring applications. In addition, this study is expected to researchers and practitioners in applying class weight techniques to other image classification tasks. With the increasing need for effective early detection systems, it is important to explore various approaches that can improve model accuracy and efficiency (Cremen & Galasso, 2020). This study can also open up opportunities for further research on the use of machine learning techniques in the environment and disaster mitigation context. We hope relevant to forest fires and can also be applied in other image classification contexts that face similar challenges. Thus, this study can contribute to global efforts in addressing the problem of forest fires and their impacts on the environment.

2. METHODS

This research uses a method framework named CRISP-DM (Cross Industry Standard Process for Data Mining). The CRISP-DM methodology provides a structured and widely accepted framework for data mining projects, comprising six key phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. It is known for offering a well-organized yet adaptable approach to conducting data-driven projects (Elkabalawy et al., 2024). Adopting the CRISP-DM methodology for this image classification of a forest fire dataset ensures that the solution developed will be robust and efficient. By following this structured framework, each phase of the project, from understanding the problem to model deployment, will be systematically addressed to provide the best results and solutions for the forest fire multiclass classification problem. The steps of our research can be seen in Figure 1.

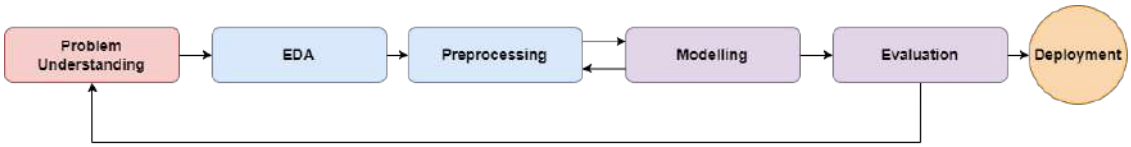


Figure 1 Research Workflow Chart

2.1 Dataset

The dataset utilized in this research was sourced from the Big Data Competition 2024 organized by the Statistics Department of Syiah Kuala University, where the data can be accessed at the Kaggle website (Bahtiar, 2024). The dataset is divided into two subsets: training and testing. The data training contains 7,331 images in .jpg format, while the data testing contains 543 images. The 7,331 images in the training set are categorized into four classes, with an uneven distribution across the categories: 3,083 images in the "None" class, 1,230 in the "Fire" class, 303 in the "Smoke" class, and 2,715 in the "Fire and Smoke" class.

Figure 2 Sample of Each Class

To facilitate model evaluation, 20% of the training data was set aside as a validation set. The class distribution within the validation set includes 619 images for the "None" class, 224 for "Fire", 67 for "Smoke", and 556 for "Fire and Smoke." This split ensures that the model is evaluated using a representative portion of the dataset across all classes.

2.2 Problem Understanding

The increasing frequency of extreme wildfires, characterized by their large scale, prolonged duration, high intensity, and severe consequences, has substantially negative effects on human health and well-being, ecosystems, the climate, and the global economy. In recent years, these extreme wildfire events have been particularly destructive (OECD, 2023). For instance, the 2015 wildfires in Indonesia resulted in an estimated economic loss of approximately USD 16 billion, equivalent to 2% of the nation's gross domestic product (GDP).

This research aims to achieve efficient and high-performing image multiclass classification. However, the dataset used in this study presents a significant challenge in reaching this goal, and that is, imbalanced data. This problem arises from an uneven distribution of target classes in the dataset, causing the model to favor the majority class while neglecting the minority classes (Bader et al., 2024a). In our case, the "fire" and "smoke" classes are the minority classes, with a significant disparity in their distribution compared to the other classes.

2.3 Data Preprocessing

Figure 3 illustrates the data preprocessing workflow using the class weighting approach. This method is based on applying class weights, which penalize the algorithm more heavily for incorrect predictions by assigning higher penalties for misclassifying the minority class (Bader et al., 2024a). Figure 4 presents the data preprocessing workflow using the data augmentation approach, which will be used to compare the performance of our primary approach, class weighting. Data augmentation methods mitigate challenges related to small datasets by artificially expanding their size and variety, ultimately improving model accuracy and generalization (Gracia Moisés et al., 2023a).

Figure 3 Class Weight Approach

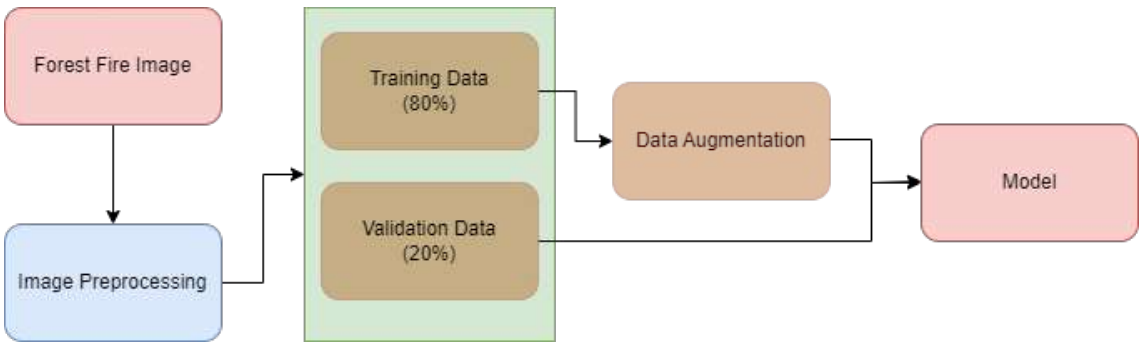


Figure 4 Data Augmentation Approach

Table 1 presents the detailed parameters and values used in the image processing steps illustrated in both approaches flow diagrams, as shown in Figures 3 and 4. Table 2 presents the detailed parameters and values of the augmentations applied to the minority classes, "fire" and "smoke," during the image transformation process in steps shown in Figure 4. The transformations included random rotations, translations, scaling, and shearing using RandomAffine, along with perspective distortions with RandomPerspective, to create new variations of the images artificially.

Table 1 Image Preprocessing Details

Parameter	Value
Resize	(224,224)
Normalize	mean=[0.485, 0.456, 0.406],

Table 2 Data Augmentation Details

Parameter	Value
RandomRotation	20
RandomPerspective	distortion_scale=0.2, p=0.5
RandomAffine	degrees=0, translate=(0.1, 0.1)

2.4 Data Augmentation

In the problem of forest fire image classification, data augmentation is a crucial technique used to address the challenge of limited and imbalanced datasets. Given the variability in environmental conditions such as lighting, smoke, and fire intensity, obtaining a sufficiently large and diverse dataset is often difficult. Data augmentation helps by artificially increasing the size of the training dataset through transformations like rotations, flips, brightness adjustments, and noise additions. These variations make the model more robust to different real-world scenarios, improving its ability to classify images, especially in challenging conditions correctly (Chlap et al., 2021).

2.5 Class Weight

Class weight is a process used to address the problem of class imbalance in image segmentation and classification tasks, where certain classes, such as those of interest, have significantly fewer examples than others. This imbalance can lead to models, especially Convolutional Neural Networks (CNNs), becoming biased towards the majority class, resulting in poor performance for the minority class. For instance, medical applications like tumor segmentation aim to make the model more sensitive to the lesion class, ensuring that critical regions, such as tumors, are accurately detected.(Ben Naceur et al., 2019).

In our forest fire classification task, where we aim to categorize images into "None," "Fire," "Smoke," and "Fire and Smoke" categories, we face a similar challenge. The "Smoke" and "Fire" classes are underrepresented compared to the "None" and "Fire and Smoke" classes, leading to an imbalance that could cause the model to underperform in detecting fire and smoke conditions, which are crucial for wildfire management. By using a class-weighting strategy, such as the one based on the Weighted Cross-Entropy function, we can assign higher weights to the minority classes (e.g., "Smoke" and "Fire"), helping the model to focus more on these underrepresented categories during training. This approach could improve the model's sensitivity toward detecting fire and smoke, much like how it effectively improved the segmentation of Glioblastoma tumors in medical applications, ensuring better detection of critical regions (Ben Naceur et al., 2019).

The balanced class weight method that we used can be explained by Equation (1) (Bakirarar & Elhan, 2023). In this formula, N represents the total number of samples in the dataset, $Class$ is the total number of unique classes, and $Sample_{class}$ is the number of samples in the respective class. By dividing the dataset size by the product of the total classes and the sample count for a given class, this method assigns higher weights to minority classes and lower weights to majority classes. This approach ensures that the training process does not disproportionately favor the majority class, improving the model's ability to generalize and recognize patterns across all classes effectively, including underrepresented ones.

$$W = \frac{N}{(class * sample)} \quad (1)$$

2.6 Modelling

Convolutional Neural Networks (CNNs) are a powerful deep learning model designed explicitly for visual data processing, such as images. Widely used for image recognition and classification tasks, CNNs excel in automatically learning spatial hierarchies of features from input data. In the study conducted by Nayak et al. (2022) a CNN-based architecture called dense EfficientNet was employed to classify 3,260 T1-weighted contrast-enhanced brain magnetic resonance images into four categories: glioma, meningioma, pituitary, and no tumor (Nayak et al., 2022). The EfficientNet architecture utilizes Inverted Residual Blocks (MBConv), similar to MobileNetV2's core building blocks. Unlike traditional CNNs, which require manual adjustments across three dimensions: depth, width, and resolution, EfficientNet employs a compound scaling approach to scale these parameters together.

Additionally, it replaces the conventional ReLU activation function with the Swish activation function, which combines linear and sigmoid components. The input image is resized to 224 × 224 to match the standard input dimensions of CNN models (Ab Wahab et al., 2021) The whole EfficientNet architecture can be seen on Figure 5.

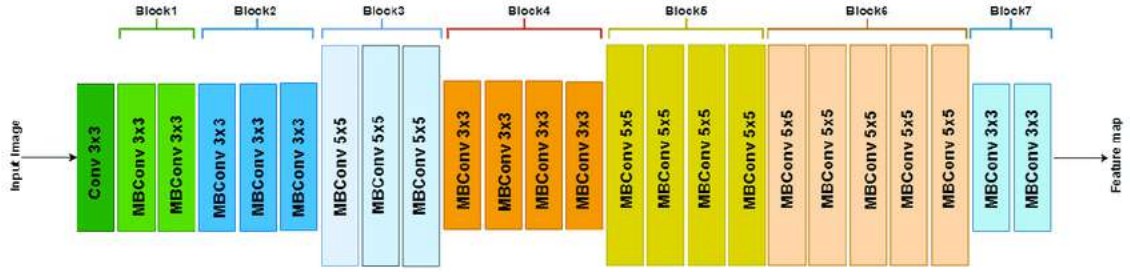


Figure 5 Architecture of EfficientNet-B1

2.7 Model Evaluation

Model evaluation is a critical process in machine learning that involves assessing the performance of a trained model using specific metrics to determine its effectiveness in making predictions. In multiclass classification, the accuracy metric is commonly used to quantify how well the model predicts the correct classes across various categories. Chen et al. (2021) highlight that maximizing training accuracy on a sufficient number of noisy samples can lead to an approximately optimal classifier even under class-conditional label noise. This finding underscores the importance of accuracy as a reliable metric for training and validation, as a noisy validation set can still provide valuable insights for model selection, including hyperparameter tuning and early stopping. By validating model performance using a noisy dataset, practitioners can achieve robust model evaluation, ensuring that the model is accurate and generalizable despite label noise (Chen et al., 2021).

This study used Accuracy (Acc) and F1-score (F1) as the primary metrics for evaluating the model. Accuracy measures the overall correctness of the model's predictions across the dataset. The F1-score, considering both precision and recall, provides a comprehensive assessment of the model's performance, especially in situations where the dataset contains uneven class distributions (Xiao et al., 2024). Accuracy (Acc) is the ratio of correctly predicted instances to the total number of instances in the dataset. In the context of multiclass classification, accuracy evaluates the percentage of correctly classified images across all classes. The equation for accuracy is given by Equation (2).

$$Accuracy = \frac{Correct\ Predictions}{Total\ Number\ of\ Predictions} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

F1-Score (F1) combines both precision (the ability of the model to identify positive instances) and recall (the ability of the model to detect all relevant positive instances). The F1-score is defined as the harmonic mean of precision and recall, making it useful for assessing the performance of minority classes. The formulas for F1-score in Equation (3), precision in Equation (4), and recall in Equation (5).

$$F1 - Score = 2\ \frac{Precision\ \ Recall}{Precision + \ Recall} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

3. RESULTS AND DISCUSSION

This section systematically outlines the model training process using the EfficientNet-B1 architecture, which was optimized to achieve high accuracy with efficient computation. The training process involved fine-tuning all layers of the model architecture to maximize accuracy. A total of 20 epochs were conducted to gather sufficient information for evaluating and optimizing the model's performance. The evaluation used a validation dataset to assess the model's ability to classify unseen images. This validation process was conducted at each epoch to monitor the model's accuracy throughout the training phase.

3.1 Training Process

Figure 6 shows the accuracy and loss throughout each epoch of the training process using the class weight method. The training accuracy consistently improves with each epoch, showing continuous growth until epoch 10. Beyond this point, up to epoch 20, there is no further improvement, indicating that the model has likely reached its peak accuracy. The model begins to stabilize in terms of validation accuracy, reaching its highest accuracy within the range of 0.88 to 0.90, starting from epoch six and continuing through epoch 20, with only slight fluctuations observed during this period.

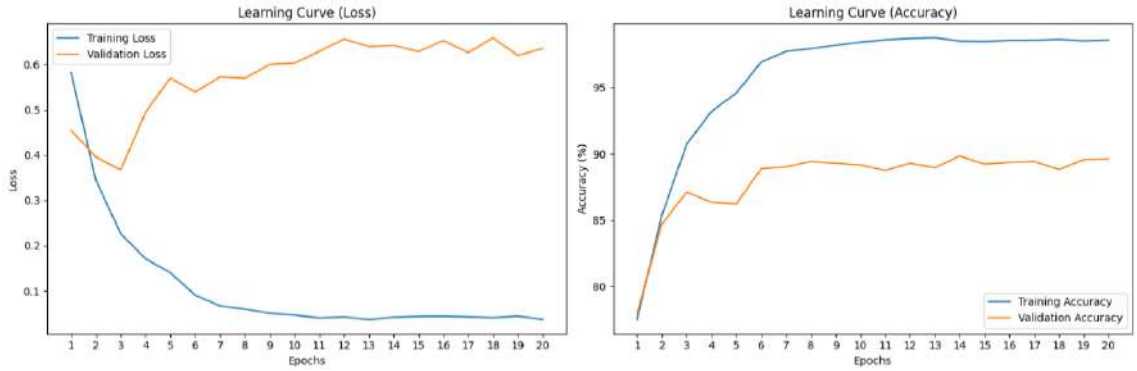


Figure 6 Loss and Accuracy of Model with Class Weight

Figure 7 shows the accuracy and loss throughout each epoch of the training process using the data augmentation method. The overall training accuracy does not differ significantly from the class weight method shown in Figure 6. However, there is a noticeable difference in the loss, with lower values ranging from 0.30 to 0.35 and smaller fluctuations. This suggests that the model, when using the augmentation method, is slightly better at addressing overfitting than the class weight method. However, the validation accuracy remains similar, consistently around 0.90.

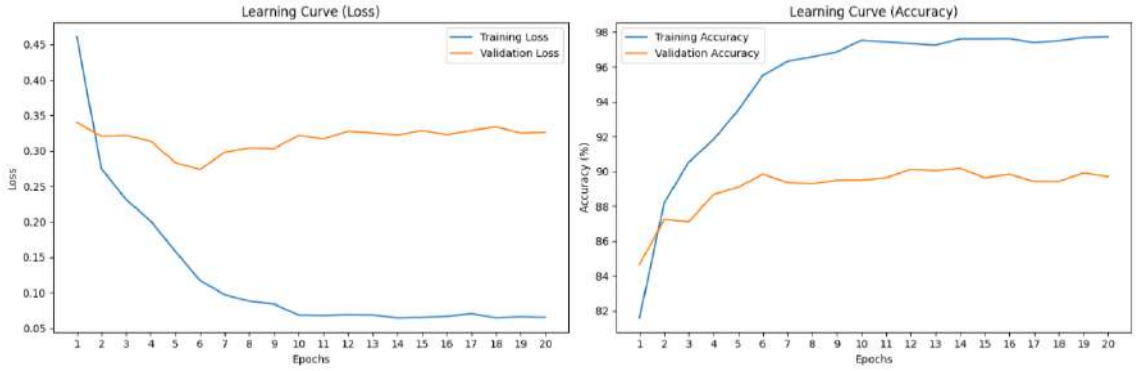


Figure 7 Loss and Accuracy of Model with Data Augmentation

However, aside fromboth methods' accuracy and loss results, the class weight method demonstrates significantly better resource and time efficiency during the training process compared to the data augmentation method. This is also quite reasonable, as the data augmentation method increases the data for the minority classes, requiring the model to train on a larger dataset. As shown in Table 3, both methods were trained using the same T4 x2 GPUs and the same model, EfficientNet-B1.

Table 3 Model Training Time with Two Different Approach

Methods	Training Time
Class Weight	14m 45s
Data Augmentation	33m 25s

3.2 Evaluation

Table 4 presents the model evaluation results using two different methods, measured by Precision, Recall, F1-Score, and Accuracy. The evaluation results are derived from the best model state based on the highest validation accuracy achieved during the training process, as shown in Figure 6 and Figure 7. The evaluation results show no significant differences between the two methods across various metrics and classes, achieving the same accuracy of 0.90. These findings show that the class weight method using the EfficientNet-B1 model on an imbalanced forest fire dataset can compete with the performance of the data augmentation approach for handling class imbalance while using around 56% less training time than the augmentation method, as shown previously in Table 3.

Table 4 EfficientNet-B1 Model Evaluation Result

	Precision		Recall		F1-Score	
	Class Weight	Data Augmentation	Class Weight	Data Augmentation	Class Weight	Data Augmentation
None	1.00	0.99	0.99	1.00	0.99	0.99
Fire	0.72	0.75	0.80	0.75	0.76	0.75
Smoke	0.72	0.81	0.76	0.72	0.74	0.76
Smoke and Fire	0.89	0.87	0.85	0.88	0.87	0.88
Accuracy					**0.90**	**0.90**

3.3 Performance Comparison with Other Models

To further assess the performance of EfficientNet-B1 in this study, we compared it to two other popular deep learning architectures, VGG-16 and ResNet50. Both models are widely used for image classification and are often benchmarked in various domains for their strong baseline

performances. Each model was trained and evaluated on the same dataset under the same configuration and method of class weighting as applied in the EfficientNet-B1 training process to ensure a fair comparison. Table 5 and Table 6 show the classification reports for VGG-16 and ResNet50, respectively, allowing for a direct comparison with the results from EfficientNet-B1 in Table 4.

Table 5 VGG-16 Model Evaluation Result

	Precision	Recall	F1-Score
None	0.98	0.98	0.98
Fire	0.52	0.79	0.63
Smoke	0.58	0.78	0.66
Smoke and Fire	0.87	0.65	0.75
Accuracy			**0.82**

Table 5 shows that VGG-16 achieves an overall accuracy of 0.82, with relatively high precision and recall for the "None" class but lower performance in distinguishing between "Fire" and "Smoke" classes. The model struggles with balancing precision and recall for "Fire" and "Smoke and Fire" categories, which are critical for accurate identification.

Table 6 ResNet50 Model Evaluation Result

	Precision	Recall	F1-Score
None	1.00	0.99	0.99
Fire	0.70	0.74	0.72
Smoke	0.76	0.76	0.76
Smoke and Fire	0.87	0.85	0.86
Accuracy			**0.89**

Table 6 indicates that ResNet50 performs better than VGG-16, with an accuracy of 0.89. ResNet50 shows high precision and recall for the "None" and "Smoke" classes, but its F1-Score for "Fire" remains slightly lower than desired. Overall, ResNet50 results better than VGG-16, especially in the "Smoke and Fire" class, where it achieves an F1-Score of 0.86.

In contrast, EfficientNet-B1 at Table 4 achieves the highest accuracy at 0.90, with consistently high precision and recall across all classes, particularly excelling in distinguishing "Fire" and "Smoke" events, with F1-Scores of 0.76 and 0.74, respectively. The balanced performance across classes, coupled with the efficient training time, suggests that EfficientNet-B1 is the best-performing model for this dataset. EfficientNet-B1's superior performance may be attributed to its optimized architecture, which balances depth, width, and resolution, making it particularly suitable for complex multiclass classification tasks like the one in this study. In conclusion, the comparison demonstrates that EfficientNet-B1 outperforms VGG-16 and ResNet50, proving the most effective model for achieving the highest accuracy in this case.

4. CONCLUSIONS

Implementing the class weight method in the EfficientNet-B1 model architecture successfully produced a successful classification model for the forest fire dataset, achieving a test accuracy of 90% within a training duration of just 14 minutes and 45 seconds. The evaluation results indicate the superiority of this class weight method in generating accurate outputs, effectively handling data imbalance during predictions, and maintaining high training efficiency. Future research could further explore alternative model architectures, such as ResNext, MobileNet, and DenseNet, to achieve even higher accuracy scores while improving computational efficiency. Additionally, future research may investigate other class weight methods beyond the balanced weight approach employed in this research, such as the Inverse of Number of Samples (INS) and Inverse of Square Root of Number of Samples (ISNS).

REFERENCES

Ab Wahab, M. N., Nazir, A., Zhen Ren, A. T., Mohd Noor, M. H., Akbar, M. F., & Mohamed, A. S. A. (2021). Efficientnet-Lite and Hybrid CNN-KNN Implementation for Facial Expression Recognition on Raspberry Pi. *IEEE Access*, *9*, 134065–134080. https://doi.org/10.1109/ACCESS.2021.3113337

Bader, M., Abdelwanis, M., Maalouf, M., & Jelinek, H. F. (2024). Detecting Depression Severity Using Weighted Random Forest and Oxidative Stress Biomarkers. *Scientific Reports*, *14*(1), 16328. https://doi.org/10.1038/s41598-024-67251-y

Bahtiar, A. (2024). *Dataset Big Data Competition USK 2024*. Kaggle. https://www.kaggle.com/datasets/arvinantobahtiar/dataset-bdc-usk-2024/data

Bakirarar, B., & Elhan, A. H. (2023). Class Weighting Technique to Deal with Imbalanced Class Problem in Machine Learning: Methodological Research. *Turkiye Klinikleri Journal of Biostatistics*, *15*(1), 19–29. https://doi.org/10.5336/biostatic.2022-93961

Barmpoutis, P., Papaioannou, P., Dimitropoulos, K., & Grammalidis, N. (2020). A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing. *Sensors*, *20*(22), 6442. https://doi.org/10.3390/s20226442

Ben Naceur, M., Kachouri, R., Akil, M., & Saouli, R. (2019). A New Online Class-Weighting Approach with Deep Neural Networks for Image Segmentation of Highly Unbalanced Glioblastoma Tumors. *International Work-Conference on Artificial Neural Networks*, *1150*, 555–567. https://doi.org/10.1007/978-3-030-20518-8_46

Benkendorf, D. J., Schwartz, S. D., Cutler, D. R., & Hawkins, C. P. (2023). Correcting for the Effects of Class Imbalance Improves the Performance of Machine-Learning Based Species Distribution Models. *Ecological Modelling*, *483*, 110414. https://doi.org/10.1016/j.ecolmodel.2023.110414

Chaturvedi, S., Khanna, P., & Ojha, A. (2022). A Survey on Vision-Based Outdoor Smoke Detection Techniques for Environmental Safety. *ISPRS Journal of Photogrammetry and Remote Sensing*, *185*, 158–187. https://doi.org/10.1016/j.isprsjprs.2022.01.013

Chen, P., Ye, J., Chen, G., Zhao, J., & Heng, P.-A. (2021). Robustness of Accuracy Metric and its Inspirations in Learning with Noisy Labels. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(13), 11451–11461. https://doi.org/10.1609/aaai.v35i13.17364

Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., & Haworth, A. (2021). A Review of Medical Image Data Augmentation Techniques for Deep Learning Applications. *Journal of Medical Imaging and Radiation Oncology*, *65*(5), 545–563. https://doi.org/10.1111/1754-9485.13261

Cremen, G., & Galasso, C. (2020). Earthquake Early Warning: Recent Advances and Perspectives. *Earth-Science Reviews*, *205*, 103184. https://doi.org/10.1016/j.earscirev.2020.103184

De Angeli, K., Gao, S., Danciu, I., Durbin, E. B., Wu, X.-C., Stroup, A., Doherty, J., Schwartz, S., Wiggins, C., Damesyn, M., Coyle, L., Penberthy, L., Tourassi, G. D., & Yoon, H.-J. (2022). Class Imbalance in Out-of-Distribution Datasets: Improving the Robustness of the TextCNN for the Classification of Rare Cancer Types. *Journal of Biomedical Informatics*, *125*, 103957. https://doi.org/10.1016/j.jbi.2021.103957

Dogra, V., Verma, S., Verma, K., Jhanjhi, N., Ghosh, U., & Le, D.-N. (2022). A Comparative Analysis of Machine Learning Models for Banking News Extraction by Multiclass Classification with Imbalanced Datasets of Financial News: Challenges and Solutions. *International Journal of Interactive Multimedia and Artificial Intelligence*, *7*(3), 35. https://doi.org/10.9781/ijimai.2022.02.002

Elkabalawy, M., Al-Sakkaf, A., Mohammed Abdelkader, E., & Alfalah, G. (2024). CRISP-DM-Based Data-Driven Approach for Building Energy Prediction Utilizing Indoor and Environmental Factors. *Sustainability*, *16*(17), 7249. https://doi.org/10.3390/su16177249

Fan, W., Si, Y., Yang, W., & Sun, M. (2022). Class-Specific Weighted Broad Learning System for Imbalanced Heartbeat Classification. *Information Sciences*, *610*, 525–548. https://doi.org/10.1016/j.ins.2022.07.074

Frederich, J., Himawan, J., & Rizkinia, M. (2024). Skin Lesion Classification Using EfficientNet B0 and B1 via TransferLearning for Computer Aided Diagnosis. *AIP Conference Proceedings*, *3080*(1), 110002. https://doi.org/10.1063/5.0200741

Gracia Moisés, A., Vitoria Pascual, I., Imas González, J. J., & Ruiz Zamarreño, C. (2023). Data Augmentation Techniques for Machine Learning Applied to Optical Spectroscopy Datasets in Agrifood Applications: A Comprehensive Review. *Sensors*, *23*(20), 8562. https://doi.org/10.3390/s23208562

Islam, Md. S. Bin, Sumon, Md. S. I., Sarmun, R., Bhuiyan, E. H., & Chowdhury, M. E. H. (2024). Classification and Segmentation of Kidney MRI Images for Chronic Kidney Disease Detection. *Computers and Electrical Engineering*, *119*, 109613. https://doi.org/10.1016/j.compeleceng.2024.109613

Madhuri, C. R., Jandhyala, S. S., Ravuri, D. M., & Babu, V. D. (2024). Accurate Classification of Forest Fires in Aerial Images Using Ensemble Model. *Bulletin of Electrical Engineering and Informatics*, *13*(4), 2650–2658. https://doi.org/10.11591/eei.v13i4.6527

Nayak, D. R., Padhy, N., Mallick, P. K., Zymbler, M., & Kumar, S. (2022). Brain Tumor Classification Using Dense Efficient-Net. *Axioms*, *11*(1), 34. https://doi.org/10.3390/axioms11010034

OECD. (2023). *Taming Wildfires in the Context of Climate Change*. OECD Publishing. https://doi.org/10.1787/dd00c367-en

Papoutsis, I., Bountos, N. I., Zavras, A., Michail, D., & Tryfonopoulos, C. (2023). Benchmarking and Scaling of Deep Learning Models for Land Cover Image Classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, *195*, 250–268. https://doi.org/10.1016/j.isprsjprs.2022.11.012

Raza, R., Zulfiqar, F., Khan, M. O., Arif, M., Alvi, A., Iftikhar, M. A., & Alam, T. (2023). Lung-EffNet: Lung Cancer Classification Using Efficientnet from CT-Scan Images. *Engineering Applications of Artificial Intelligence*, *126*, 106902. https://doi.org/10.1016/j.engappai.2023.106902

Rodríguez, J. J., Díez-Pastor, J.-F., Arnaiz-González, Á., & Kuncheva, L. I. (2020). Random Balance Ensembles for Multiclass Imbalance Learning. *Knowledge-Based Systems*, *193*, 105434. https://doi.org/10.1016/j.knosys.2019.105434

Tanase, M. A., Aponte, C., Mermoz, S., Bouvet, A., Le Toan, T., & Heurich, M. (2018). Detection of Windthrows and Insect Outbreaks by L-Band SAR: A Case Study in the Bavarian Forest National Park. *Remote Sensing of Environment*, *209*, 700–711. https://doi.org/10.1016/j.rse.2018.03.009

Tanveer, M., Sharma, A., & Suganthan, P. N. (2021). Least Squares KNN-Based Weighted Multiclass Twin SVM. *Neurocomputing*, *459*, 454–464. https://doi.org/10.1016/j.neucom.2020.02.132

Xiao, Y., Zhao, J., Yu, Y., Ding, X., Liu, S., Bao, W., Wen, S., & Zhou, X. (2024). SimpleCNN-UNet: An Optic Disc Image Segmentation Network Based on Efficient Small-Kernel Convolutions. *Expert Systems with Applications*, *256*, 124935. https://doi.org/10.1016/j.eswa.2024.124935

Zhao, J., Jin, J., Chen, S., Zhang, R., Yu, B., & Liu, Q. (2020). A Weighted Hybrid Ensemble Method for Classifying Imbalanced Data. *Knowledge-Based Systems*, *203*, 106087. https://doi.org/10.1016/j.knosys.2020.106087

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 74 – 86
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Application of SMOTE in Sentiment Analysis of MyXL User Reviews on Google Play Store

Badriyah [(1)*], Totok Chamidy [(2)], Suhartono [(3)]
Department of Informatics, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, Indonesia
e-mail : 200605220010@student.uin-malang.ac.id, {to2k2013,suhartono}@ti.uin-malang.ac.id.
* Corresponding author.
This article was submitted on 30 July 2024, revised on 26 September 2024, accepted on 26 September 2024, and published on 31 January 2025.

Abstract

Texts that express customer opinions about a product are important input for companies. Companies obtain valuable information from consumer perceptions of marketed products by conducting sentiment analysis. However, real-world text datasets are often unbalanced, causing the prediction results of classification algorithms to be biased towards the majority class and ignore the minority class. This study analyzes the sentiment of MyXL user reviews on the Google Play Store by comparing the performance of the Logistic Regression and Support Vector Machine algorithms in the SMOTE implementation. This analysis uses TF-IDF to extract feature and GridSearchCV to optimize the accuracy, precision, recall, and F1 score evaluation metrics. This study follows several scenarios of dividing training data and test data. SVM implementing SMOTE is the algorithm with the best performance using the division of training data (90%) and test data (10%), resulting in accuracy (73.00%), precision (67.13%), recall (65.82%) and F1 score (66.30%).

Keywords: Sentiment Analysis, Logistic Regression, Support Vector Machine, GridSearchCV, SMOTE

Abstrak

Teks yang mengungkapkan opini pelanggan tentang suatu produk merupakan masukan penting bagi perusahaan. Perusahaan memperoleh informasi berharga dari persepsi konsumen terhadap produk yang dipasarkan dengan melakukan analisis sentimen. Namun, kumpulan data teks dunia nyata seringkali tidak seimbang sehingga menyebabkan hasil prediksi algoritma klasifikasi menjadi bias terhadap kelas mayoritas dan mengabaikan kelas minoritas. Penelitian ini menganalisis sentimen ulasan pengguna MyXL di Google Play Store dengan membandingkan kinerja algoritma Logistic Regression dan Support Vector Machine pada implementasi SMOTE. Analisis ini menggunakan TF-IDF untuk ekstraksi fitur dan GridSearchCV untuk mengoptimalkan metrik evaluasi akurasi, presisi, recall, dan skor F1. Penelitian ini mengikuti beberapa skenario pembagian data latih dan data uji. SVM yang mengimplementasikan SMOTE merupakan algoritma dengan performa terbaik dengan menggunakan pembagian data latih (90%) dan data uji (10%), menghasilkan akurasi (73,00%), presisi (67,13%), recall (65,82%) dan skor F1 (66,30%)**.**

Kata Kunci: Analisis Sentimen, *Logistic Regression*, *Support Vector Machine*, GridSearchCV, SMOTE

1. INTRODUCTION

Customer opinions are one of the main indicators for evaluating a product's success. In a highly competitive world, listening to the voice of customers is crucial to gaining deeper insights into what consumers truly desire and how they respond to changes made by companies. Sentiment analysis allows businesses to understand users' perceptions of their products or services. By analyzing sentiment, companies can identify strengths and weaknesses from the customer's perspective. This identification helps in improving services and developing products to better align with market needs. Additionally, customer opinions serve as valuable information for other customers (Hasibuan & Heriyanto, 2022). A survey of over 7,000 consumers across 11 Asia-Pacific regions revealed that 76% of consumers seek reviews to validate a company before

making a purchase (Cheng & Mani, 2024). Customer experiences significantly influence their purchasing decisions. Companies must actively respond to both positive and negative reviews professionally and promptly, demonstrating that they value customer feedback and are willing to address shortcomings.

Customers express opinions in the form of text, commonly found on social media, online marketplaces, and applications in the Google Play Store. These text data serve as input for machine learning algorithms to analyze and classify sentiment as positive, negative, or neutral. In real-world conditions, datasets often exhibit imbalances in the distribution of data among classes. This imbalance greatly affects the accuracy and reliability of sentiment analysis results, as the classification tends to be biased toward the majority class. For instance, if most training data consists of positive sentiment, the model is likely to be more accurate in detecting positive sentiment but less accurate in identifying neutral or negative sentiment. Such data imbalance causes the model to ignore or misclassify minority class predictions (Khushi et al., 2021). Hence, research is needed to develop and evaluate methods capable of effectively handling data imbalance, ensuring accurate predictions across all classes.

SMOTE is frequently used for sentiment analysis in datasets with imbalanced class distributions. For instance, sentiment analysis of Twitter data about IndihomeCare used SMOTE alongside Support Vector Machine (SVM), AdaBoost, and Particle Swarm Optimization algorithms (Syah et al., 2023). The dataset comprised 1,000 records, with 653 positive reviews and 570 negative reviews. In this study, the application of SMOTE and SVM achieved the highest evaluation scores, proving effective for the given dataset.

Similarly, sentiment analysis of public opinions about antibiotic use in Indonesia employed SVM (Darwis et al., 2023). Out of 1,889 tweets collected through web scraping, 1,631 were negative sentiments, and 258 were positive sentiments. This study implemented SVM with linear, RBF, and polynomial kernels, using RoBERTa-based labeling, cross-validation training, and bigram tokenization methods. Three different text preprocessing scenarios were tested, including TF-IDF feature extraction and SMOTE for addressing class imbalance. The results demonstrated a significant improvement in SVM performance after applying SMOTE.

Another example is sentiment analysis of netizen opinions on various international bag brands, utilizing SMOTE for classification model optimization (Huda et al., 2023). The cleaned dataset from Twitter reviews contained 2,881 reviews, including 1,083 positive, 374 negative, and 1,424 neutral sentiments. This study compared the performance of several algorithms, including Logistic Regression, Multinomial Naïve Bayes, Decision Tree, K-Nearest Neighbors, Random Forest, and SVM. SVM achieved the best performance with an accuracy of 69%. After applying SMOTE, the SVM model's accuracy improved to 82%.

A sentiment analysis study on the metaverse compared Naïve Bayes and Logistic Regression algorithms using SMOTE optimization (Ramadhani & Suryono, 2024). This research analyzed 6,728 comments about the metaverse on the X (formerly Twitter) social media platform using a text mining approach. The optimization results showed that Logistic Regression outperformed Naïve Bayes, achieving a higher accuracy of 95% compared to 91%. The studies mentioned and the planned research are summarized in Table 1.

Logistic Regression and SVM are highly popular algorithms for text classification and are widely used in sentiment analysis. Both algorithms were originally designed for binary classification tasks but have been developed to handle multiclass classification effectively. This study aims to compare the performance of these two algorithms in applying SMOTE for sentiment analysis using an imbalanced multiclass dataset. The novelty of this research lies in comparing the performance of Logistic Regression and SVM using SMOTE, TF-IDF, and GridSearchCV hyperparameter tuning in sentiment analysis of user reviews of the MyXL application on Google Play Store.

Table 1 Related Research

No.	Researchers	Title	Research Features	Research Plan
1	Syah et al. (2023)	Sentiment Analysis of IndihomeCare Twitter Using Comparison of SMOTE, Support Vector Machine, and AdaBoost Algorithms	- Comparison of: 1. SMOTE, SVM 2. SMOTE, SVM, and AdaBoost 3. SMOTE, Particle Swarm Optimization - Twitter Crawling - RapidMiner	- Comparing SMOTE application on Logistic Regression and SVM - GridSearchCV - MyXL user reviews on Google Play Store - Python
2	Darwis et al. (2023)	Support Vector Machine for Public Sentiment Analysis on Antibiotic Use in Indonesia	- Comparing SMOTE and non-SMOTE on SVM with linear, RBF, and polynomial kernels - Preprocessing comparison: slang words by Pujangga and Ramaprokoso, stopwords by NLTK and Sastrawi - TF-IDF - Twitter Crawling - RapidMiner	- Comparing SMOTE and non-SMOTE application on Logistic Regression and SVM - TF-IDF - GridSearchCV - MyXL user review dataset from Google Play Store - Python
3	Huda et al. (2023)	Optimization of Netizen Sentiment Classification Model for Foreign Brand Bags	- Comparing SMOTE application on Logistic Regression, Multinomial Naïve Bayes, Decision Tree, KNN, Random Forest, and SVM - SMOTE applied to the entire dataset before train-test split - TF - Twitter Crawling	- Comparing SMOTE and non-SMOTE application on Logistic Regression and SVM - SMOTE on training data - TF-IDF - GridSearchCV - MyXL user review dataset from Google Play Store
4	Ramadhani & Suryono (2024)	Comparison of Naïve Bayes and Logistic Regression Algorithms for Sentiment Analysis of the Metaverse	- Comparing SMOTE and non-SMOTE on Naïve Bayes and Logistic Regression - SMOTE applied to the entire dataset before train-test split - TF-IDF - Crawling X (formerly Twitter)	- Comparing SMOTE and non-SMOTE application on Logistic Regression and SVM - SMOTE on training data - TF-IDF - GridSearchCV - MyXL user review dataset from Google Play Store

2. METHODS

2.1 Dataset Preparation

The object of this study is the user review dataset for the MyXL application on Google Play Store, comprising 1,000 data points. The dataset contains 613 negative sentiment reviews, 226 neutral sentiment reviews, and 161 positive sentiment reviews. The imbalanced distribution of classes in this dataset is ideal for testing machine learning algorithms designed to handle imbalanced data. The MyXL user review dataset can be downloaded from Kaggle (Audiansyah, 2022). An example of user review data is presented in Table 2.

Table 2 Sample Data of MyXL User Reviews on Google Play Store

Data	Review	Sentiment
Review 1	Tolong dong masalah jaringan hampir setiap hari leg parahh!! Jangan buat kecewa costumer lah	Negative
Review 2	..... kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama. terima kasih	Positive
Review 3	Bisa cek kuota dgn simpel terbaikkkkkkkkkkkk!!	Positive

2.2 System Design

The system design for this study, as shown in Figure 1, begins with preparing the MyXL user review dataset. Next, the dataset undergoes text preprocessing and keyword weighting using TF-IDF. For model testing, the dataset is split into training and testing datasets. SMOTE oversampling is applied to balance class distributions in the training data. The Logistic Regression and Support Vector Machine classification algorithms are developed with hyperparameter tuning using the GridSearchCV object from the Scikit-learn library. Predictions are made on the testing data, and evaluation is conducted by comparing the predictions against the actual classes.

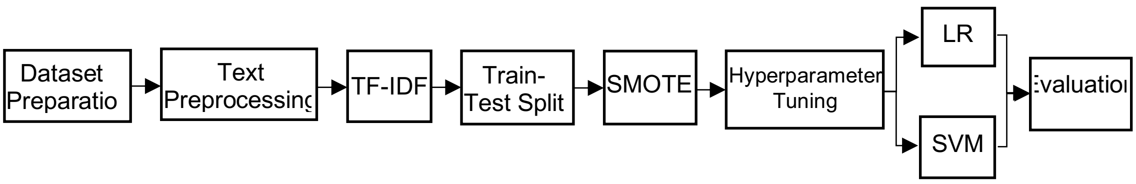


Figure 1 Research System Design

2.3 Text Preprocessing

Text preprocessing involves a series of techniques to transform raw text data into a format suitable for processing and analysis (Haikal et al., 2024). The MyXL user review dataset from Google Play Store, as unstructured raw data, contains various writing styles and language variations. To enable machine learning on this data, it must be converted into a structured format through preprocessing. Table 3 displays examples of MyXL user reviews that have been preprocessed using these techniques.

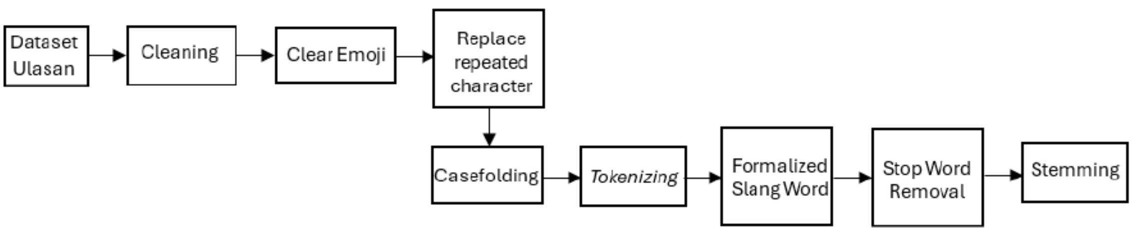


Figure 2 Text Preprocessing Steps

Table 3 Sample Results of Text Preprocessing

Stage	Review 1	Review 2	Review 3
Original Review	Tolong dong masalah jaringan hampir setiap hari leg parahh!! Jangan buat kecewa costumer lah	..... kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama. terima kasih	Bisa cek kuota dgn simpelterbaikkkkkkkkkk kk!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!
Cleaning	Tolong dong masalah jaringan hampir setiap hari leg parahh Jangan buat kecewa costumer lah	. kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama terima kasih	Tolong dong masalah jaringan hampir setiap hari leg parahh Jangan buat kecewa costumer lah
Clear Emoji	Tolong dong masalah jaringan hampir setiap hari leg parahh Jangan buat kecewa costumer lah	kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama terima kasih	Bisa cek kuota dgn simpel terbaikkkkkkkkkk
Replace Repeated Characters	Tolong dong masalah jaringan hampir setiap hari leg parahh Jangan buat kecewa costumer lah	kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama terima kasih	Bisa cek kuota dgn simpel terbaik
Casefolding	tolong dong masalah jaringan hampir setiap hari leg parahh jangan buat kecewa costumer lah	kalau bisa masa aktif kartunya di tingkatkan jadi lebih lama terima kasih	bisa cek kuota dgn simpel terbaik
Tokenizing	'tolong', 'dong', 'masalah', 'jaringan', 'hampir', 'setiap', 'hari', 'leg', 'parahh', 'jangan', 'buat', 'kecewa', 'costumer', 'lah'	'kalau', 'bisa', 'masa', 'aktif', 'kartunya', 'di', 'tingkatkan', 'jadi', 'lebih', 'lama', 'terima', 'kasih'	'bisa', 'cek', 'kuota', 'dgn', 'simpel', 'terbaik'
Formalizing Slang Words	'tolong', 'dong', 'masalah', 'jaringan', 'hampir', 'setiap', 'hari', 'lelet', 'parah', 'jangan', 'buat', 'kecewa', 'konsumen', 'lah'	'kalau', 'bisa', 'masa', 'aktif', 'kartunya', 'di', 'tingkatkan', 'jadi', 'lebih', 'lama', 'terima', 'kasih'	'bisa', 'cek', 'kuota', 'dengan', 'simpel', 'terbaik'
Removing Stopwords	'tolong', 'jaringan', 'lelet', 'parah', 'kecewa', 'konsumen'	'aktif', 'kartunya', 'tingkatkan', 'terima', 'kasih'	'cek', 'kuota', 'simpel', 'terbaik'
Stemming	'tolong', 'jaring', 'lelet', 'parah', 'kecewa', 'konsumen'	'aktif', 'kartu', 'tingkat', 'terima', 'kasih'	'cek', 'kuota', 'simpel', 'baik'

The text preprocessing steps, illustrated in Figure 2, include:

a) Text cleaning: Removing noise such as mentions, hashtags, numbers, and specific characters, replacing them with spaces, and trimming leading or trailing spaces.
b) Emoji removal: Removing emojis from the text.
c) Character reduction: Eliminating repeated characters that appear three times or more.
d) Case folding: Converting all text to lowercase to simplify the text features.
e) Tokenization: Splitting text into tokens (words) using the Natural Language Toolkit (NLTK) library.
f) Slang formalization: Replacing slang terms with formal equivalents using a predefined slang dictionary stored in a .txt file.

g) Stopword removal: Eliminating Indonesian stopwords listed in the NLTK library.
h) Stemming: Converting words to their root forms using the Sastrawi stemmer, with the Swifter library speeding up DataFrame operations in pandas.

2.4 TF_IDF and Train-Test Split

The MyXL user review dataset consists of text data in string format, which cannot be directly processed by machine learning algorithms. It must first be transformed into numerical or vector representations, a process known as feature extraction. In this study, TF-IDF (Term Frequency-Inverse Document Frequency) was used for keyword extraction. TF-IDF calculates the weight of a term in a document by considering its frequency and importance across the entire corpus (Febrianti et al., 2023).

a) **TF (Term Frequency):** Measures how often a term appears in a document (tf_{ij} is the frequency of term i in document j).
b) **IDF (Inverse Document Frequency):** Assesses the significance of a term in the corpus, as shown in Equation (1), where N is the total number of documents, and df_i is the number of documents containing term i.
c) **TF-IDF:** A combination of TF and IDF, obtained by multiplying them, as shown in Equation (2).

$$idf_i = \log\left(\frac{N}{df_i}\right) + 1 \quad (1)$$

$$\begin{aligned} w_{ij} &= tf_{ij} * idf_i \\ &= tf_{ij} * (\log\left(\frac{N}{df_i}\right) + 1) \end{aligned} \quad (2)$$

The MyXL user review dataset contains 1,000 records and 1,589 features. Table 4 presents the average TF-IDF values for all features, including “aamiin” (0.2507), “abal” (0.8652), and “yutube” (0.3249). The train-test split is a simple, commonly used validation method that divides the dataset into training and testing sets. This division is necessary for training and evaluating the algorithm. The numerical dataset from feature extraction is typically split using ratios of 90:10, 80:20, or 70:30. The split is done randomly while maintaining class proportions in both sets, mirroring the original class distribution. Table 5 shows the training and testing data counts for a 90:10 split.

Table 4 Average TF-IDF Values of All Features

Index	0	1	2	3	…	1,855	1,856	1,857	1,858
Feature	Aamiin	abal	abang	abg	…	yt	yth	yuk	yutube
TF-IDF	0.2507	0.8652	0.1933	0.3206	…	1.1781	0.4507	0.5644	0.3249

Table 5 Data Splitting Using a 90:10 Ratio

Review Category	Negative Class	Neutral Class	Positive Class	Total	%
Training Data	552	203	145	900	90
Testing Data	61	23	16	100	10

2.5 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is a widely used oversampling technique that synthesizes new samples for minority classes to increase their representation before classification. SMOTE works by selecting a sample from the minority class and identifying its k-nearest neighbors. Synthetic samples are then generated along the line segments connecting the original sample to its neighbors, based on the required level of oversampling (Chawla et al., 2002).

SMOTE aims to address class imbalance during model training by balancing the class distributions in the training data, allowing the model to learn the minority class patterns effectively. As such, SMOTE is applied only to the training data, leaving the testing data imbalanced to reflect real-world conditions. Testing on imbalanced data provides valid, objective, and accurate model performance evaluation. Applying SMOTE to the entire dataset before splitting would introduce data leakage, as the model would have access to synthetic patterns from the testing data during training, invalidating the evaluation. Table 6 displays the training data distribution after SMOTE application.

Table 6 Number of Data After Applying SMOTE on Training Data

Review Category	Negative Class	Neutral Class	Positive Class	Total
Before SMOTE	552	203	145	900
	61.3%	22.6%	16.1%	100%
After SMOTE	552	552	552	1,656
	33.3%	33.3%	33.3%	100%

2.6 Hyperparameter Tuning

Hyperparameters are parameters that control the learning process of a machine learning algorithm (Nishat et al., 2022). Hyperparameter tuning involves adjusting these parameters to find the optimal combination for maximizing model performance. This study employed GridSearchCV to train the algorithm and identify the best model by exploring all possible hyperparameter combinations.

GridSearchCV performs cross-validation by dividing the training data into 10 subsets. The model is trained on nine subsets and validated on one, with the process repeated until each subset has served as a validation set. This approach optimizes hyperparameter tuning, identifying the best parameters and achieving the highest cross-validation score. For Logistic Regression, the hyperparameters include C and penalty values, while for SVM, they include C, gamma, and kernel.

2.7 Logistic Regression

Logistic Regression predicts the relationship between independent variables and categorical dependent variables, which may be either nominal or ordinal. For datasets where the dependent variable is nominal with more than two categories, Multinomial Logistic Regression is used (Harahap et al., 2023).

The formula for Multinomial Logistic Regression is expressed in Equation (3). It predicts the probability of a particular observation i belonging to a given class in a dataset. In this formula, $\pi(X_i)$ represents the estimated probability of the i-th observation, which is calculated based on the independent variables associated with that observation. The equation incorporates β_0, which is the constant or intercept term that adjusts the baseline probability for all observations. Additionally, β_k denotes the coefficient for the k-th independent variable, which measures the influence of that variable on the predicted probability. X_{ik} represents the value of the k-th independent variable for the i-th observation. Together, these components define the relationship between the independent variables and the estimated probabilities, enabling classification into multiple categories.

$$\pi(X_i) = \frac{\exp\left(\beta_0 + \sum_{k=1}^{n} \beta_k X_{ik}\right)}{1 + \exp\left(\beta_0 + \sum_{k=1}^{n} \beta_k X_{ik}\right)} \quad (3)$$

2.8 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm that uses a linear function hypothesis in high-dimensional spaces, trained through optimization algorithms that apply

learning biases derived from statistical theory (Ovirianti et al., 2022). The primary goal of SVM is to create an optimal separating function that can be used for classification tasks.

SVM's basic principle is linear classification, initially limited to handling binary class problems. However, its capabilities have been enhanced through the kernel concept, allowing it to address non-linear problems and multiclass classification. Important parameters in the SVM algorithm include the penalty (L) and the kernel (Atmanegara & Purwa, 2021).

The equations for the **linear** and **polynomial kernels** in Support Vector Machine (SVM) help define the transformation of input data into a higher-dimensional feature space where it becomes easier to separate classes. In these equations, x_i and x_j represent the dot product of two feature vectors, which quantifies their similarity in the feature space. The parameter γ acts as a scale control, commonly set to $1/number\ of\ features$, and influences the flexibility of the decision boundary. The parameter r serves as a bias term, adjusting the output of the kernel function to improve fit. For the polynomial kernel, an additional parameter d specifies the degree of the polynomial, where higher degrees allow for more complex decision boundaries. These components collectively enable SVM to adapt to both linear and non-linear classification tasks by adjusting how input data is mapped and classified in the transformed feature space.

$$K_{linear}(x_{i,}x_j) = x_i^T\ x_j \tag{4}$$

$$K_{polynomial}(x_{i,}x_j) = (\gamma x_i^T\ x_j + r)^d, \gamma > 0 \tag{4}$$

2.9 Research Scenarios

Table 7 Research Scenarios

Scenario	Training Data (%)	Testing Data (%)	Training Data Balance	Algorithm
1	90	10	Before SMOTE	Logistic Regression
			Before SMOTE	Support Vector Machine
			After SMOTE	Logistic Regression
			After SMOTE	Support Vector Machine
2	80	20	Before SMOTE	Logistic Regression
			Before SMOTE	Support Vector Machine
			After SMOTE	Logistic Regression
			After SMOTE	Support Vector Machine
3	70	30	Before SMOTE	Logistic Regression
			Before SMOTE	Support Vector Machine
			After SMOTE	Logistic Regression
			After SMOTE	Support Vector Machine
4	60	40	Before SMOTE	Logistic Regression
			Before SMOTE	Support Vector Machine
			After SMOTE	Logistic Regression
			After SMOTE	Support Vector Machine
5	50	50	Before SMOTE	Logistic Regression
			Before SMOTE	Support Vector Machine
			After SMOTE	Logistic Regression
			After SMOTE	Support Vector Machine

This study evaluates the performance of Logistic Regression and SVM, employing SMOTE to address data imbalance in sentiment analysis of MyXL reviews. It involves hyperparameter tuning with GridSearchCV across five experimental scenarios, as shown in Table 7.

2.10 Evaluasi

Evaluation measures classification accuracy to assess algorithm performance. Multiclass sentiment classification accuracy is calculated using metrics such as True Positive (TP), False Negative (FN), True Negative (TN), and False Positive (FP) for each class C_i. Overall accuracy is calculated as a macro average of all classes, providing an unbiased performance summary across classes (Grandini et al., 2020). The equations for macro-average metrics are shown in Table 8.

Table 8 Formula for Macro Average Classification Accuracy

Metric	Formula
Accuracy rata-rata	$\frac{\sum_{i=1}^{n} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{n}$
Precision	$\frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i + fp_i}}{n}$
Recall	$\frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i + fn_i}}{n}$
F score	$2 * \frac{Precision * Recall}{Precision + Recall}$

3. RESULTS AND DISCUSSION

Hyperparameter tuning using GridSearchCV for training Logistic Regression (LR) and Support Vector Machine (SVM) produced the parameter combinations shown in Table 9. These parameters resulted in the best sentiment classification models for user reviews of the MyXL application on Google Play Store. Multiclass classification accuracy for the scenarios is detailed in Table 10. From Table 10, Scenario 1 achieved the highest accuracy across all models before applying SMOTE: LR (71.00%) and SVM (70.00%). The same SVM accuracy (70.00%) was also observed in Scenario 4. However, accuracy metrics in imbalanced datasets may not fully reflect the model's true performance. For a more comprehensive evaluation, the F1-score, a harmonic mean of precision and recall, must be considered. A higher F1-score indicates better precision-recall balance. In Scenario 1, SVM achieved a higher F1-score (65.10%) than its accuracy (62.75%), highlighting that SVM Scenario 1 outperformed SVM Scenario 4. Furthermore, Scenario 1 yielded the highest accuracy and F1-score for all models after applying SMOTE, with LR accuracy at 72.00%, SVM accuracy at 73.00%, LR F1-score at 63.94%, and SVM F1-score at 66.30%. This makes Scenario 1 the most effective configuration for all algorithms compared to other scenarios. The classification accuracy results for all scenarios are visualized in Figure 3.

In Scenario 1, Logistic Regression accuracy after SMOTE (72.00%) surpassed its pre-SMOTE accuracy (71.00%). Other metrics showed similar improvements: precision increased from 62.95% to 64.12%, recall from 63.83% to 64.38%, and F1-score from 63.01% to 63.94%. The increases—accuracy (+1%), precision (+1.17%), recall (+0.55%), and F1-score (+0.93%)—demonstrate that SMOTE consistently improved Logistic Regression's performance. These improvements indicate that the model became better at identifying patterns in minority classes and achieving balanced classifications after data distribution was equalized using SMOTE.

In Scenario 1, SVM accuracy after SMOTE (73.00%) exceeded its pre-SMOTE accuracy (70.00%). Precision increased from 64.44% to 67.13%, and F1-score rose from 65.10% to 66.30%. Accuracy improved by 3%, precision by 2.69%, and F1-score by 1.2%. These substantial improvements show that SMOTE significantly enhanced SVM's ability to classify data accurately and reduce false positives. However, SVM's recall decreased slightly after SMOTE, from 65.99% to 65.82%, a minor drop of 0.17%. Despite this, the F1-score increase (1.2%) demonstrates that SMOTE effectively improved SVM's overall performance in handling imbalanced data.

Table 9 Hyperparameter Tuning GridSearchCV Result

	Algorithm	Before SMOTE	After SMOTE
Scenario 1	LR	C = 1,1263157894736844	C = 4,0
		Penalty = L2	Penalty = L2
	SVM	C = 1,0	C = 1,0
		Gamma = 0,2	Gamma = scale
		Kernel = linear	Kernel = poly
Scenario 2	LR	C = 1,9473684210526316	C= 3,3842105263157896
		Penalty = L2	Penalty = L2
	SVM	C = 1,0	C = 1,0
		Gamma = scale	Gamma = scale
		Kernel = poly	Kernel = poly
Scenario 3	LR	C = 0,5105263157894737	C = 4,0
		Penalty = L2	Penalty = L1
	SVM	C = 1,0	C = 1,0
		Gamma = 0,2	Gamma = scale
		Kernel = linear	Kernel = poly
Scenario 4	LR	C = 1,3315789473684212	C = 3,3842105263157896
		Penalty = L2	Penalty = L2
	SVM	C = 1,0	C = 1,0
		Gamma = 0,2	Gamma = scale
		Kernel = linear	Kernel = poly
Scenario 5	LR	C = 3,3842105263157896	C = 4,0
		Penalty = L2	Penalty = L2
	SVM	C = 1,0	C = 1,0
		Gamma = 0.2	Gamma = scale
		Kernel = linear	Kernel = poly

Table 10 Macro Average Classification Accuracy of LR and SVM

	Algorithm	Before SMOTE				After SMOTE			
		Acc (%)	Prec (%)	Rec (%)	F1 (%)	Acc (%)	Prec (%)	Rec (%)	F1 (%)
Scenario 1	LR	71,00	62,95	63,83	63,01	72,00	64,12	64,38	63,94
	SVM	70,00	64,44	65,99	65,10	73,00	67,13	65,82	66,30
Scenario 2	LR	69,50	61,49	62,61	61,72	69,50	61,74	62,61	62,00
	SVM	69,00	61,31	63,75	62,34	68,50	60,33	60,36	60,27
Scenario 3	LR	66,67	58,64	59,67	59,12	67,33	58,19	59,61	58,71
	SVM	66,00	60,03	61,05	60,42	68,67	60,69	60,65	60,63
Scenario 4	LR	69,50	60,46	60,58	60,34	69,50	60,54	61,20	60,44
	SVM	70,00	62,98	62,54	62,75	69,25	62,50	60,37	61,30
Scenario 5	LR	67,20	58,26	57,99	58,10	68,20	59,70	58,78	59,13
	SVM	66,00	59,49	58,77	58,98	67,00	60,06	57,21	58,39

In Scenario 1, before SMOTE, SVM had slightly lower accuracy (70.00%) than LR (71.00%), but SVM outperformed LR in precision (64.44%), recall (65.99%), and F1-score (65.10%). This indicates that SVM was better at identifying minority classes before SMOTE. After SMOTE, SVM surpassed LR across all metrics, demonstrating that SMOTE enabled SVM to more effectively identify and classify minority classes. The evaluation results for Scenario 1 are visualized in Figure 4.

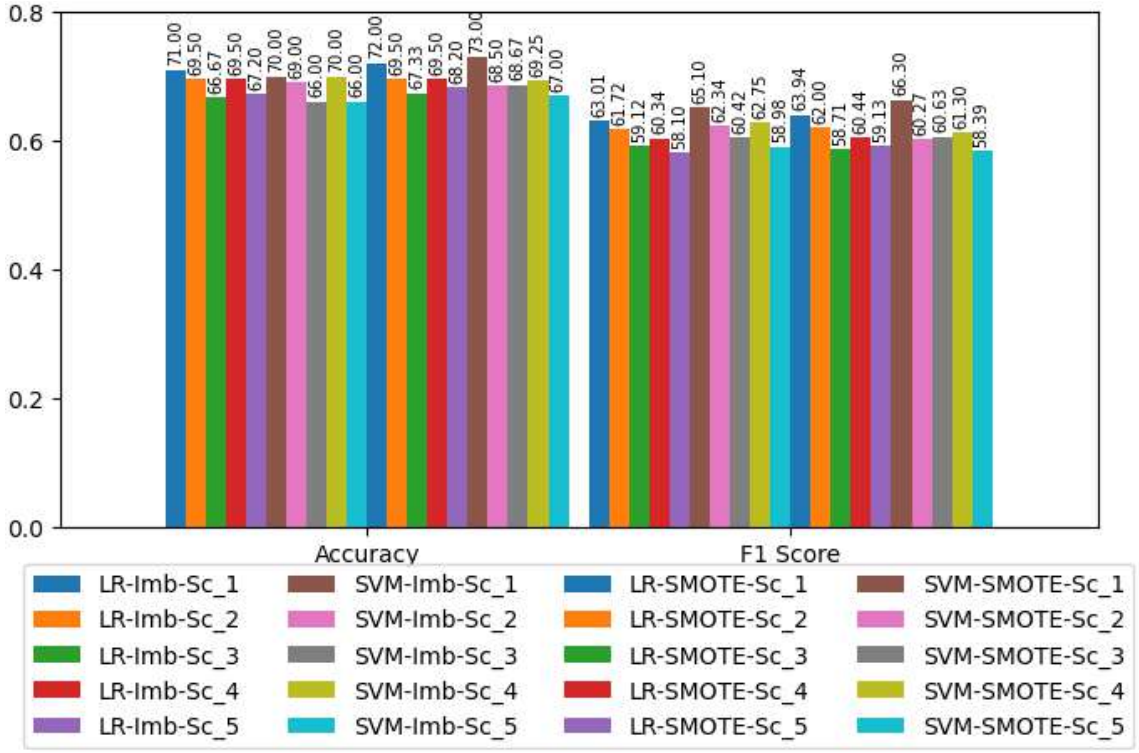


Figure 3 Evaluation Results of All Research Scenarios

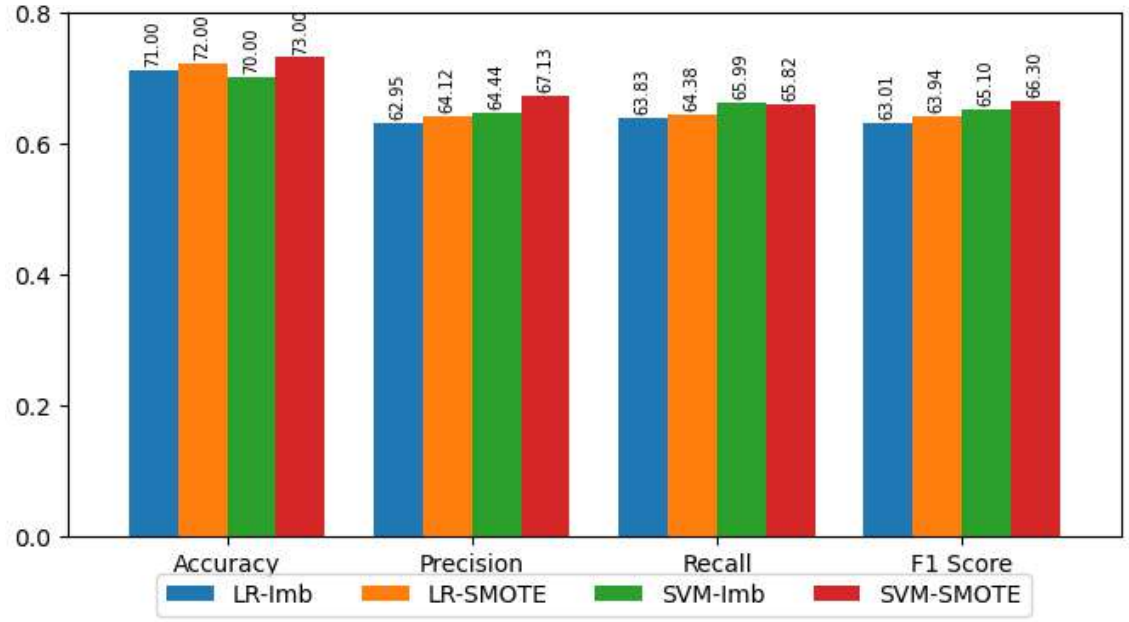


Figure 4 Comparison of Evaluation Results in Scenario 1

Post-SMOTE, the SVM algorithm achieved the highest classification accuracy with a 90% training and 10% testing split, using the parameter combination $C = 1.0$, $\gamma = scale$, and $kernel = poly$. Predictions on the test data produced a confusion matrix shown in Table 11. From this, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values for each class were calculated and are presented in Table 12. Table 13 displays the overall multiclass classification accuracy, calculated using the macro-average formula from Table 8.

Table 11 Confusion Matrix of SVM After SMOTE in Scenario 1

Actual Class	Predicted Class		
	Negative	Neutral	Positive
Negative	52	7	2
Neutral	10	10	3
Positive	3	2	11

Table 12 Classification Accuracy per Class for SVM After SMOTE in Scenario 1

Class	TP	FP	TN	FN
Negative	52	13	26	9
Neutral	10	9	68	13
Positive	11	5	79	5

Table 13 Macro Average Classification Accuracy of SVM After SMOTE in Scenario 1

Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
73,00	67,13	65,82	66,30

4. CONCLUSIONS

This study compares the performance of Logistic Regression (LR) and Support Vector Machine (SVM) algorithms using SMOTE, TF-IDF, and GridSearchCV for sentiment analysis on the MyXL user review dataset from Google Play Store. The GridSearchCV object from the Scikit-learn library played a crucial role in hyperparameter tuning for both algorithms. The parameter combinations yielding the best models were applied to the algorithms for evaluation. The 90% training and 10% testing data split demonstrated the highest performance for both LR and SVM models, both before and after applying SMOTE. In the context of imbalanced data, SVM outperformed LR in identifying minority classes. Applying SMOTE enhanced the performance of both algorithms, with SVM continuing to show superior capabilities in recognizing and classifying minority classes.

The SVM algorithm achieved the best performance using SMOTE with parameter combinations $C = 1.0$, $\gamma = scale$, and $kernel = poly$, resulting in a classification accuracy of 73.00%, precision of 67.13%, recall of 65.82%, and F1-score of 66.30%. The lack of significant improvement in evaluations before and after SMOTE might stem from the nature of SMOTE, which performs well in certain cases but does not always produce substantial improvements in all situations. To address this, more in-depth hyperparameter tuning is necessary after applying SMOTE. While GridSearchCV explores all possible hyperparameter combinations, making it computationally intensive, RandomizedSearchCV can serve as an alternative. This method conducts random searches within a large parameter space, offering greater efficiency in terms of time.

REFERENCES

Atmanegara, E., & Purwa, T. (2021). Hybrid Support Vector Machine and Logistic Regression for Multiclass Classification: A Case Study on Wine Dataset. *Indonesian Journal of Data Science*, *1*(1), 1–7. https://www.researchgate.net/publication/353211298

Audiansyah, D. D. (2022, July 5). *Data Ulasan Terlabel*. Kaggle. https://www.kaggle.com/datasets/dimasdiandraa/data-ulasan-terlabel?select=Ulasan+My+XL+1000+Data+Labelled.csv

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Cheng, M., & Mani, R. (2024, June 24). *Voice of the Consumer Survey 2024: Asia Pacific*. PWC Indonesia. https://www.pwc.com/id/en/pwc-publications/industries-publications/consumer-and-industrial-products-and-services/consumer-survey-2024-asia-pacific.html

Darwis, H., Wanaspati, N., & Anraeni, S. (2023). Support Vector Machine untuk Analisis Sentimen Masyarakat terhadap Penggunaan Antibiotik di Indonesia. *The Indonesian Journal of Computer Science*, *12*(4), 12. https://doi.org/10.33022/ijcs.v12i4.3320

Febrianti, F. A. D. P., Hamami, F., & Fa'rifah, R. Y. (2023). Aspect-Based Sentiment Analysis terhadap Ulasan Aplikasi Flip Menggunakan Pembobotan Term Frequency-Inverse Document Frequency (TF-IDF) dengan Metode Klasifikasi K-Nearest Neighbors (K-NN). *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, *4*(3), 1858–1873. https://doi.org/10.35870/jimik.v4i3.429

Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: An Overview*. http://arxiv.org/abs/2008.05756

Haikal, M., Martanto, M., & Hayati, U. (2024). Analisis Sentimen terhadap Penggunaan Aplikasi Game Online PUBG Mobile Menggunakan Algoritma Naive Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(6), 3275–3281. https://doi.org/10.36040/jati.v7i6.8174

Harahap, F. H., Sutarman, S., Darnius, O., & Sembiring, P. (2023). Klasifikasi Menggunakan Model Regresi Logistik Multinomial dan Regresi Logistik Multinomial Komponen Utama. *IJM: Indonesian Journal of Multidisciplinary*, *1*(2), 632–642. https://journal.csspublishing.com/index.php/ijm/article/view/183

Hasibuan, E., & Heriyanto, E. A. (2022). Analisis Sentimen pada Ulasan Aplikasi Amazon Shopping di Google Play Store Menggunakan Naive Bayes Classifier. *Jurnal Teknik dan Science*, *1*(3), 13–24. https://doi.org/10.56127/jts.v1i3.434

Huda, M. N., Fauzan, D. A., Pamungkas, M. R. S. P., Ratnadewi, N. S., & Vahendra, A. A. (2023). Optimalisasi Model Klasifikasi Sentimen Netizen terhadap Merek Tas Luar Negeri. *Jurnal KomtekInfo*, 21–28. https://doi.org/10.35134/komtekinfo.v10i1.360

Khushi, M., Shaukat, K., Alam, T. M., Hameed, I. A., Uddin, S., Luo, S., Yang, X., & Reyes, M. C. (2021). A Comparative Performance Analysis of Data Resampling Methods on Imbalance Medical Data. *IEEE Access*, *9*, 109960–109975. https://doi.org/10.1109/ACCESS.2021.3102399

Nishat, M. M., Faisal, F., Ratul, I. J., Al-Monsur, A., Ar-Rafi, A. M., Nasrullah, S. M., Reza, M. T., & Khan, M. R. H. (2022). A Comprehensive Investigation of the Performances of Different Machine Learning Classifiers with SMOTE-ENN Oversampling Technique and Hyperparameter Optimization for Imbalanced Heart Failure Dataset. *Scientific Programming*, *2022*, 1–17. https://doi.org/10.1155/2022/3649406

Ovirianti, N. H., Zarlis, M., & Mawengkang, H. (2022). Support Vector Machine Using A Classification Algorithm. *Jurnal dan Penelitian Teknik Informatika*, *6*(3). https://doi.org/10.33395/sinkron.v7i3

Ramadhani, B., & Suryono, R. R. (2024). Komparasi Algoritma Naïve Bayes dan Logistic Regression untuk Analisis Sentimen Metaverse. *Jurnal Media Informatika Budidarma*, *8*(2), 714. https://doi.org/10.30865/mib.v8i2.7458

Syah, F., Fajrin, H., Afif, A. N., Saeputra, M. R., Mirranty, D., & Saputra, D. D. (2023). Analisa Sentimen terhadap Twitter IndihomeCare Menggunakan Perbandingan Algoritma Smote, Support Vector Machine, AdaBoost dan Particle Swarm Optimization. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, *7*(1), 53–58. https://doi.org/10.35870/jtik.v7i1.686

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 87 – 99
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Revitalizing Art with Technology: A Deep Learning Approach to Virtual Restoration

Nurrohmah Endah Putranti [(1)], Shyang-Jye Chang [(2)], Muhammad Raffiudin [(3)*]
[1,2] Department of Mechanical Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan
[3] Department of Computer Science, Chiang Mai University, Chiang Mai, Thailand
e-mail : {m11111073,changjye}@yuntech.edu.tw, muhammad_r@cmu.ac.th.
* Corresponding author.
This article was submitted on 20 October 2024, revised on 7 November 2024, accepted on 7 November 2024, and published on 31 January 2025.

Abstract

This study evaluates CycleGAN's performance in virtual painting restoration, focusing on color restoration and detail reproduction. We compiled datasets categorized by art styles and conditions to achieve accurate restorations without altering original reference materials. Various paintings and those with a yellow filter, to create effective training datasets for CycleGAN. The model utilized cycle consistency loss and advanced data augmentation techniques. We assessed the results using PSNR, SSIM, and Color Inspector metrics, focusing on Claude Monet's Nasturtiums in a Blue Vase and Hermann Corrodi's Prayers at Dawn. The findings demonstrate superior color recovery and preservation of intricate details compared to other methods, confirmed through quantitative and qualitative evaluations. Key contributions include employing CycleGAN for art restoration, model evaluation, and framework development. Practical implications extend to art conservation, digital library enhancement, art education, and broader access to restored works. Future research may explore dataset expansion, complex architectures, interdisciplinary collaboration, automated evaluation tools, and improved technologies for real-time restoration applications. In conclusion, CycleGAN holds promise for digital art conservation, with ongoing efforts aimed at integrating across fields for effective cultural preservation.

Keywords: Art Restoration, CycleGAN, Deep Learning, PSNR, SSIM

Abstrak

Penelitian ini mengevaluasi kinerja CycleGAN dalam restorasi lukisan virtual, dengan fokus pada pemulihan warna dan detail. Kami menyusun dataset yang dikategorikan berdasarkan gaya seni dan kondisi untuk mencapai restorasi akurat tanpa mengubah materi referensi asli. Beberapa lukisan, termasuk yang dengan filter kuning, dirusak untuk membuat dataset pelatihan yang efektif bagi CycleGAN. Model ini memanfaatkan cycle consistency loss dan teknik augmentasi data canggih. Hasil dievaluasi menggunakan metrik PSNR, SSIM, dan Color Inspector, dengan fokus pada *Nasturtiums in a Blue Vase* karya Claude Monet dan *Prayers at Dawn* karya Hermann Corrodi. Temuan menunjukkan pemulihan warna yang unggul dan pelestarian detail halus dibandingkan metode lain, yang dikonfirmasi melalui evaluasi kuantitatif dan kualitatif. Kontribusi utama termasuk penggunaan CycleGAN untuk restorasi seni, evaluasi model, dan pengembangan kerangka kerja. Implikasi praktis mencakup konservasi seni, peningkatan perpustakaan digital, pendidikan seni, dan akses yang lebih luas terhadap karya yang direstorasi. Penelitian di masa depan dapat mengeksplorasi perluasan dataset, arsitektur kompleks, kolaborasi lintas disiplin, alat evaluasi otomatis, dan teknologi untuk aplikasi restorasi waktu nyata. Kesimpulannya, CycleGAN menunjukkan potensi dalam konservasi seni digital, dengan upaya terus berlanjut untuk integrasi dengan bidang lain guna pelestarian budaya yang efektif.

Kata Kunci: Restorasi Seni, CycleGAN, *Deep Learning*, PSNR, SSIM

1. INTRODUCTION

Preserving the intrinsic value of artwork is a primary goal of art restoration. Art collectors often assign lower value to paintings with damage, such as discoloration, dullness, yellowing, and other aesthetic flaws, compared to those in good condition. Over time, a painting's varnish layer can accumulate dirt and yellow, altering its appearance. Removing the varnish can restore the original hues, but physical restoration poses risks, especially for delicate or valuable works. Traditional painting restoration uses techniques like solvent cleaning and varnish removal, though some methods may not be fully reversible and could permanently affect the artwork. Virtual restoration offers a safer alternative, allowing experimentation without damaging the original piece. Deep Learning techniques, like CNNs (J.-Y. Zhu et al., 2017) and GANs (Kumar & Gupta, 2024), have shown significant potential in image-processing tasks, including virtual art restoration.

Virtual restoration involves using digital technology and software to restore and preserve artwork in a virtual environment. This process employs various techniques to repair and enhance the visual appearance of paintings without physically altering the originals. The virtual restoration method enables non-destructive testing and reversible adjustments, offering a safer approach for experimentation. Previous works used CNNs to restore yellow-filtered images to their original colors (Maali Amiri & Messinger, 2023). The research focuses on color restoration because it is one of a painting's most visually and artistically significant aspects. Accurate color restoration ensures the painting's aesthetic is preserved and the artist's original vision is maintained.

Building on prior GAN model (Wu et al., 2021), we propose using CycleGAN, which can handle unpaired data, to clean the varnish layer virtually and restore the artwork's colors. CycleGAN automatically learns to translate images between two collections, enabling effective virtual restoration. In this work, we attempt to achieve virtual restoration of artworks, specifically paintings, using deep learning, with CycleGAN as the network architecture and enhance the appearance of varnished paintings that have undergone yellowish degradation. The underlying near-original colors can be revealed by virtually removing the varnish, mitigating the yellowing effect, and restoring the artwork's authentic color tones. CycleGAN is particularly effective for tasks where paired training data is scarce or unavailable, often in art restoration. Cycle Consistency Loss is a unique feature of CycleGAN that plays a crucial role in ensuring the transformation process is reversible, thereby maintaining the integrity of the original image. This loss function helps the model learn to translate an image from one domain to another and back again, ensuring that important features and details are preserved throughout the translation process. CycleGAN has been successfully applied to style transfer tasks, which are closely related to the challenges faced in virtual painting restoration.

1.1 Related Work

Virtual restoration offers insight into how a painting appeared, reviving its colors to align with the artist's intent. It ensures the long-term preservation and accessibility of restoration data. Virtual restoration uses digital technology to enhance and preserve artworks without physically altering them. This approach employs non-destructive, reversible techniques, making it a safer option for experimentation. Additionally, virtual restoration is accessible to a broader audience, including researchers, artists, and enthusiasts, through appropriate software and resources (Pietroni & Ferdani, 2021).

Integrating Machine Learning in art restoration has transformed the field, enabling automated, efficient, and high-quality processes. Several approaches have been proposed: Farajzadeh & Hashemzadeh (2021) used U-Net and CNN for digital inpainting of damaged Persian pottery, focusing on noise removal, brightness adjustment, and sharpening. Sizyakin et al. (2022) employed U-Net to detect cracks in murals and GAN for inpainting the cracks. And J. Wang et al. (2023) introduced a gradient-guided dual-branch GAN to generate high-quality relic sketches. These methods highlight the versatility of machine learning techniques in addressing various restoration challenges, although they often require paired data or specific conditions for optimal performance.

H.-L. Wang et al. (2018) propose a systematic restoration framework for high-resolution deteriorated mural textures that is both efficient and effective. By using patches cropped from the original texture as training data, they achieved successful restoration results, as demonstrated in the Dunhuang mural restoration of the 61st cave. Similarly, Maali Amiri & Messinger (2021) achieved a more accurate and generalizable virtual cleaning method for paintings by using a convolutional neural network (CNN). Their approach outperforms existing physical models in restoring color quality and spectral similarity, successfully applying to famous artworks like the Mona Lisa without prior detailed information. Zou et al. developed a virtual restoration method for weathered paintings on ancient Chinese buildings using multiple deep-learning algorithms. Their approach segments the painting into the background, golden edges, and dragon patterns, applying different restoration techniques. The result provides a layered restoration that aids traditional restorers in visualizing the artwork's original appearance, reducing repetitive work and complexity (Zou et al., 2021).

J.-Y. Zhu et al. (2017) introduced an approach for image-to-image translation without the need for paired training data. Their model, CycleGAN, learns to map images from one domain to another using adversarial loss while ensuring cycle consistency through inverse mappings. This approach was tested on tasks like style transfer and photo enhancement, showing qualitative and quantitative improvements over previous methods. Engin et al. (2018) enhanced this concept with Cycle-Dehaze, an end-to-end network for single-image dehazing that combines cycle consistency and perceptual losses to improve texture recovery. Xiao et al. (2019) presented a CycleGAN-based colorization method for single grayscale images, introducing high-level semantic identity loss and low-level color loss for better optimization.

Wan et al. (2020) introduced a deep-learning method to restore old photos with severe degradation. Traditional supervised learning approaches struggle due to the complex degradation in real photos and the domain gap between synthetic and real images. To overcome this, they developed a triplet domain translation network using real photos and synthetic image pairs. They bridge the domain gap by training two VAEs to map old and clean photos into latent spaces. Their method also incorporates a global branch for structured defects (e.g., scratches) and a local branch for unstructured defects (e.g., noise), leading to superior restoration performance compared to state-of-the-art methods.

Despite these advancements in virtual restoration and machine learning applications, a research gap remains in developing models that can effectively address the unique challenges of virtual painting restoration, particularly in terms of color fidelity and detail preservation. While many existing methods focus on specific aspects of restoration or rely on paired data, there is a need for a more comprehensive approach that combines the strengths of various techniques. This research aims to fill these gaps by leveraging CycleGAN's capabilities in unpaired image translation and cycle consistency to enhance color restoration in paintings. The objectives of this study are to evaluate CycleGAN's effectiveness in maintaining color fidelity and to explore its potential for broader application in virtual art restoration.

2. METHODS

2.1 Deep Learning in Image Processing

CycleGAN (Cycle-Consistent Generative Adversarial Network) is a generative adversarial network (GAN) designed explicitly for image-to-image translation tasks where paired training data is unavailable. Introduced by J.-Y. Zhu et al. (2017), CycleGAN learns to translate images between domains without needing paired examples, making it particularly effective for applications such as style transfer, object transfiguration, and domain adaptation. Two key data augmentation techniques were employed to effectively prepare the dataset for training the CycleGAN model to enhance its diversity and realism: yellow filtering for varnished paintings and image resizing. Yellow filtering was applied to simulate the aging effect commonly seen in varnished artworks, resulting in a yellowish or sepia-toned hue for the images.

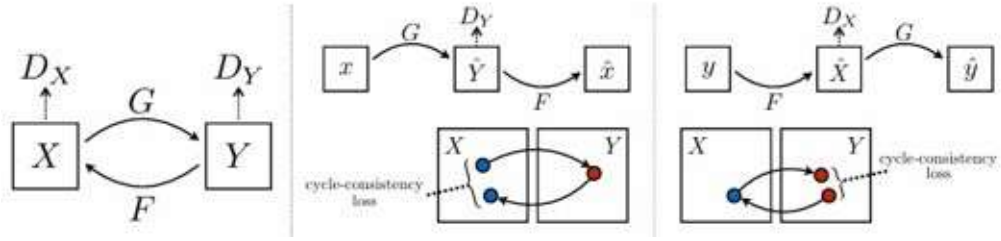


Figure 1 Concept CycleGAN

The learning rate is a critical hyperparameter that determines the step size at each iteration toward minimizing the loss function. In CycleGAN models, a common learning rate is 0.0002, which helps ensure stable training. Learning rate decay is often employed, gradually reducing the learning rate after a specified number of epochs, such as a linear decay to zero over the final 100 epochs. This gradual reduction allows the model to converge more smoothly by making smaller weight updates as training progresses. The batch size refers to the number of training examples used in a single iteration. For CycleGANs, smaller batch sizes, such as 1 or 2, are often employed. These smaller sizes are preferred because they fit better in GPU memory and are beneficial for training GANs, which can be sensitive to mini-batch statistics. Additionally, smaller batch sizes can contribute to more stable training dynamics.

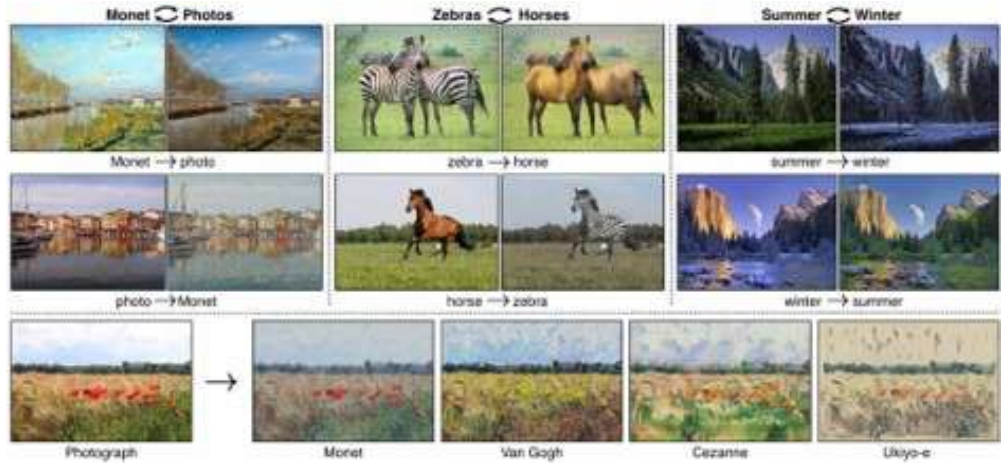


Figure 2 Example of Application CycleGAN (Maali Amiri & Messinger, 2023)

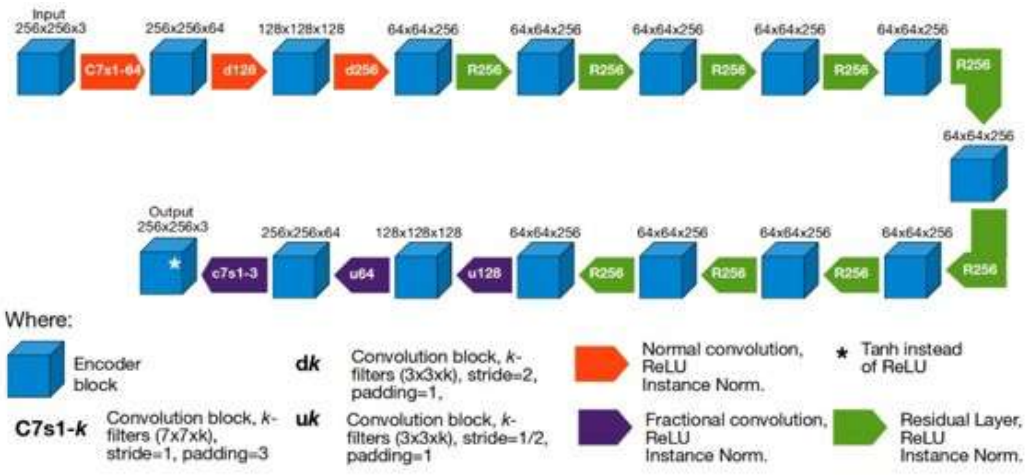


Figure 3 Internal Architecture CycleGAN

Color calibration involves adjusting and correcting the colors of an image or display to ensure they align with a standard or desired outcome. This process guarantees that the colors in the final image are accurate and consistent across different devices and media. The primary goal of color calibration is to achieve true-to-life color reproduction, minimizing discrepancies due to variations in devices, lighting conditions, and environmental factors.

Incorporating Macbeth palette images into the training dataset enhances the color calibration by providing a standardized reference that improves color accuracy. The Macbeth ColorChecker palette, featuring 24 color patches representing natural objects, is a reliable standard for adjusting colors in the restored images. This ensures that the output images remain consistent with known color values, thereby preserving the original hues and tones of the paintings (Maali Amiri & Messinger, 2023).

Figure 4 Macbeth Color-Checker Palette (1 – 4 filtered as varnished reference, Color-Checker as original/unvarnished reference)

2.2 Experimental Dataset and Parameter Environment

For this study, the dataset used in the virtual restoration process was collected from several reputable online sources. These websites provided high-quality digital images of various artworks, which were selected based on specific criteria such as degradation, varnish yellowing, and visual impairments commonly addressed in restoration. The collected dataset was then preprocessed and curated to ensure diversity in art styles and conditions, allowing for a comprehensive evaluation of the proposed CycleGAN model. This curated dataset played a crucial role in training and validating the model for effective virtual restoration. However, to further enhance the model's reliability, it is important to provide more quantity and variety of art styles and degradation conditions within the dataset. Adding more data variety, including abstract styles or other types of damage, would help strengthen the model's performance. Additionally, plans or suggestions for increasing data variety in future work should be considered, as they would contribute to a more robust and versatile model capable of handling a more comprehensive range of restoration challenges.

Several techniques augment the dataset, such as random cropping, which extracts patches from images to increase diversity and reduce overfitting. Scaling and resizing images to a fixed size (256x256 pixels) standardizes input dimensions for efficient training. Additionally, techniques like horizontal and vertical flipping and color jittering (random adjustments to brightness, contrast, saturation, and hue) introduce variations for the model to learn.

Figure 5 The result of the resize step

2.3 Testing and Performance Metrics

PSNR (Peak Signal-to-Noise Ratio) offers a clear numerical representation of the difference between the restored and reference images. It is a valuable metric for assessing the overall quality and fidelity of the restoration process, as higher PSNR values indicate more accurate restorations with less distortion or noise. SSIM (Structural Similarity Index Measure) assesses the perceptual quality of images by considering changes in structural information essential for human visual perception. This makes SSIM especially relevant for artistic image restoration tasks, where preserving structural integrity is vital for a faithful representation of the original artwork.

PSNR quantifies the ratio between the maximum possible power of a signal (image) and the power of corrupting noise that degrades its fidelity. This metric is expressed in decibels (dB) and is commonly used to assess the overall quality of image restoration processes.

$$PNSR = 10.log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (1)$$

MAX represents the image's maximum pixel value (typically 255 for 8-bit images). At the same time, MSE (Mean Squared Error) is the average of the squared differences between corresponding pixels of the restored and reference images.

A higher PSNR value indicates better image quality, suggesting lower distortion or noise in the restored image compared to the reference image. Because of its ability to quantify the fidelity of reconstructed images, PSNR is widely used in image restoration tasks. Euclidean distance measures the straight-line distance between two points in multi-dimensional space. In image restoration, it quantifies the pixel-wise differences between the restored and reference images, indicating how closely the restoration matches the original.

2.4 Implementation

Training CycleGAN poses several challenges, with mode collapse being a significant issue, where the generator produces limited output diversity. This can be mitigated by adding noise to inputs

or using different mini-batches to update the generators and discriminators. Balancing adversarial and cycle consistency losses is crucial for stability, and adjusting the weights λcyc and λidentity can aid in achieving this balance. Techniques such as gradient clipping can help prevent exploding gradients, and using instance normalization instead of batch normalization enhances training stability and performance. Finally, due to the resource-intensive nature of CycleGAN training, employing strategies like mixed-precision training and distributed training across multiple GPUs can significantly accelerate the process.

In this work, the CycleGAN model was configured with a batch size of 1, a learning rate 0.0002, and cycle loss lambda values of 10. The model was trained for 1000 epochs, with regular checkpoints saved to monitor progress and prevent overfitting. The batch size is set to 1, allowing the model to update its parameters based on each image pair's gradient information. This configuration helps manage memory constraints while facilitating fine-grained updates. Additionally, with Cycle Consistency Loss set to LAMBDA_CYCLE = 10, the model ensures consistency in transformations between varnished and unvarnished images.

CycleGAN's optimization employs Adam optimizers for the generators and discriminators, leveraging its effectiveness with large datasets and an adaptive learning rate mechanism. Typical parameters for the Adam optimizer include β1 = 0.5, β2 = 0.999, and a learning rate of 0.0002, striking a balance between convergence speed and stability. Additionally, a linear learning rate decay is applied after a specified number of epochs (e.g., after the first 100 epochs in a 200-epoch training cycle), aiding in fine-tuning the model towards the end of training.

3. RESULTS AND DISCUSSION

3.1 Qualitative Results

The results showcase the visual results of the CycleGAN model applied to the restoration of varnished paintings, demonstrating its effectiveness in removing and restoring the original appearance. Notable examples include Monet's "Nasturtiums in a Blue Vase" and Hermann Corrodi's "Prayer at Dawn," which feature different styles and restoration challenges, highlighting the model's robustness and versatility. Figure 6 features a still life by Claude Monet, where the varnish has dulled the vibrancy of the flowers and background. The original varnished image displays a significant yellow tint that obscures the vivid colors of the nasturtiums and blue vase. In contrast, the restored image reveals Monet's intended true colors, showcasing the flowers' vibrancy and the blue vase's prominence.

Figure 6 Original Unvarnish Traditionally (left), Virtual Unvarnish Restoration (middle), Original Varnished Painting (right)

Figure 7 features a historical artwork by Hermann Corrodi depicting a serene dawn prayer scene, affected by varnish that has dimmed its overall brightness. The original image appears muted, with a yellowish haze reducing the clarity and vibrancy of the scene. In contrast, the restored image reveals richer, more accurate colors, enhancing the prominence and clarity of the dawn

light and architectural details. This comparative analysis underscores the strengths of the CycleGAN model in virtual painting restoration. While the lack of direct visual comparisons is limited, the model's quantitative performance and qualitative improvements are promising. Future work should focus on comprehensive visual comparisons and expanding datasets to validate our findings further.

Figure 7 Original Unvarnish Traditionally (left), Virtual Unvarnish Restoration (middle), Original Varnished Painting (right)

The comparison of paintings before and after virtual restoration reveals significant improvements in several key areas. One of the most noticeable enhancements is the improved color accuracy. The restoration model effectively eliminates the yellow tint caused by varnish, thereby restoring the original hues intended by the artist. This correction not only revives the aesthetic appeal of the artwork but also ensures that the artist's original vision is preserved. In addition to color restoration, the model demonstrates proficiency in recovering fine details previously obscured by the varnish. This capability is crucial for maintaining the integrity and authenticity of the artwork, as it allows viewers to appreciate the intricate details and textures that define the painting.

Furthermore, the restoration often produces a cleaner image with reduced noise, offering a clearer artwork view. This noise reduction enhances the overall visual quality and allows a more accurate interpretation of the painting's features and elements. Together, these improvements underscore the effectiveness of the restoration model in enhancing both the aesthetic and historical value of artworks.

3.2 Quantitative Results

This section evaluates the performance of the CycleGAN model using various quantitative metrics, which provide a numerical assessment of the model's ability to restore varnished paintings. The metrics employed include Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Euclidean Distance, and Color Inspector 3D. These metrics collectively offer insights into the model's effectiveness in enhancing image quality, preserving structural integrity, and accurately restoring colors. By analyzing these metrics, we can objectively assess the CycleGAN model's performance in virtual painting restoration.

PSNR (Peak Signal-to-Noise Ratio) measures the ratio of the maximum possible signal power to the power of corrupting noise, quantifying image quality by comparing the original unvarnished image to the restored image. SSIM (Structural Similarity Index Measure) evaluates image quality based on structural information, luminance, and contrast between the original and restored images, with values ranging from -1 to 1; higher values indicate greater structural similarity. Euclidean Distance calculates the straight-line distance between corresponding pixels in the RGB color space of the original and restored images, with lower distances indicating higher color fidelity. The 3D Color Inspector provides a visual representation of the color distribution in the

original and restored images within 3D RGB space, allowing for visual comparison to identify discrepancies.

Several aspects are considered to evaluate the quality of the results. First, the Average Euclidean Distance is assessed, where a low value suggests that the two images are very similar, while a high value indicates significant color differences. Second, the Distance Distribution is analyzed; a graph with most points near the Y-axis, indicating small distance values, suggests that many pixels are very similar in both images, whereas a wide distribution implies large variations, indicating potential differences between the images. Third, Peaks in the Graph are examined; high peaks at low distance values indicate that many pixels are almost identical in the two images, while scattered peaks suggest variations in pixel similarity. Finally, PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index Measure) are considered, with a high PSNR typically indicating very similar images and an SSIM value close to 1, signifying a high degree of structural similarity. These aspects collectively provide a comprehensive assessment of image quality and similarity.

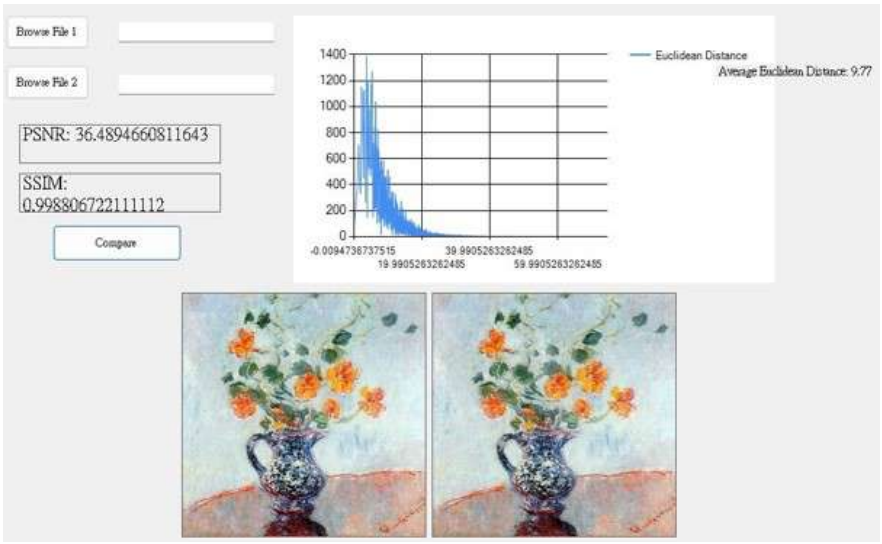


Figure 8 Result of Monet painting analysis PNSR, SSIM, and Euclidean Distance

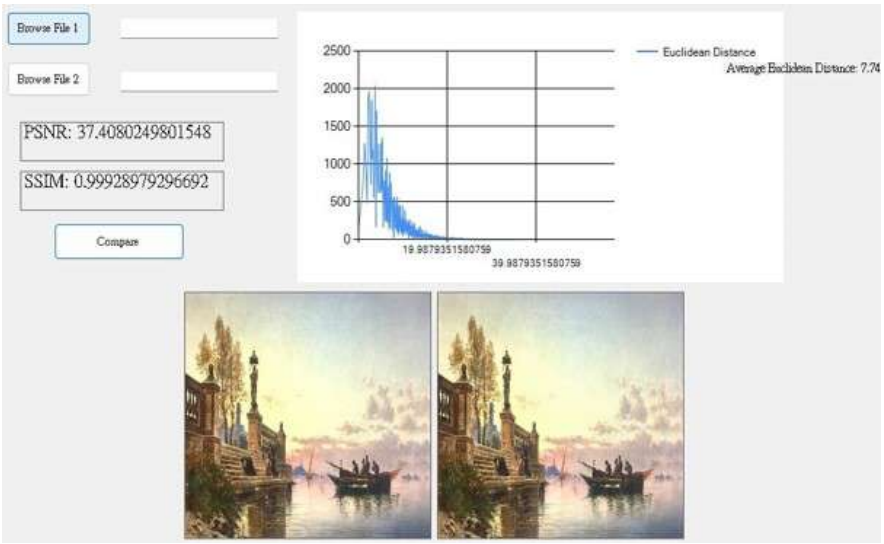


Figure 9 The result of Hermann painting analysis is PNSR, SSIM, and Euclidean Distance

Understanding the example results involves analyzing image similarity through specific metrics. As shown in Figure 8 and Figure 9, highly similar images are indicated by points on the graph that are close to the Y-axis, reflecting a high Peak Signal-to-Noise Ratio (PSNR), typically above 30 dB, and a high Structural Similarity Index Measure (SSIM), often above 0.9. These metrics suggest superior image quality and strong similarity between images. Conversely, dissimilar images exhibit a wide distribution of points on the graph, with low PSNR values, typically below 20 dB, and low SSIM scores, often below 0.5. PSNR values serve as image quality indicators, with higher values denoting better quality. SSIM assesses the similarity between two images based on luminance, contrast, and structure, ranging from -1 to 1, where 1 signifies perfect similarity. These metrics collectively provide a comprehensive understanding of image quality and similarity.

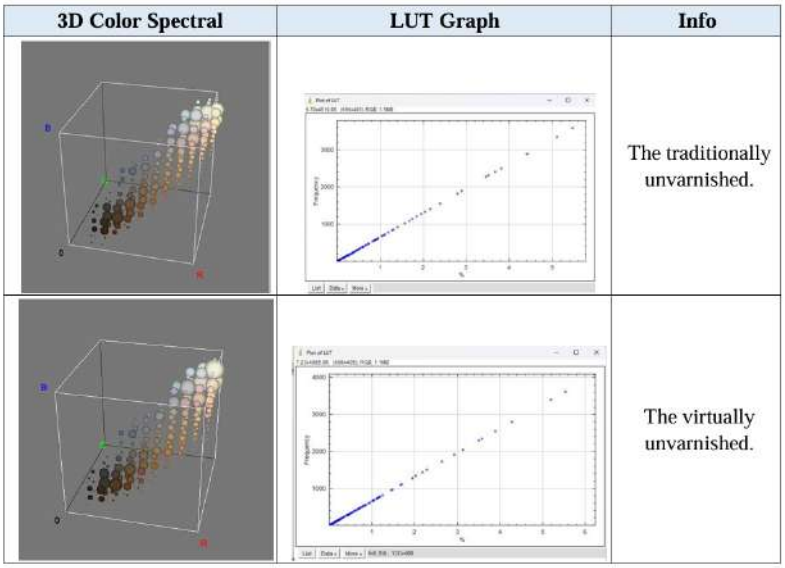


Figure 10 Corrodi - "Prayer at Dawn" Color Inspector 3D Simulation

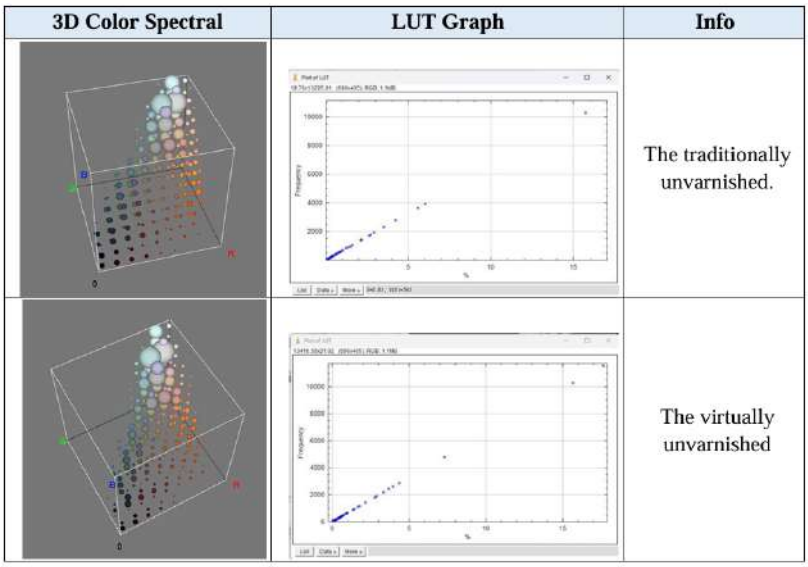


Figure 11 Monet - "Nasturtiums in a Blue Vase" Color Inspector 3D Simulation

Figures 10 and 11 utilize Color Inspector 3D to visually and quantitatively evaluate the quality of color restoration. Consistent color distributions in these figures indicate successful restoration

efforts. The Look-Up Table (LUT) graph is particularly useful, as it displays how colors in the input image are mapped to those in the output (restored) image. By including the LUT graph for this painting, specific observations can be discussed, such as areas where color consistency is well maintained and any noticeable shifts that may occur. Additionally, the 3D Cube RGB spectral color feature in Color Inspector 3D is an advanced tool for visualizing and analyzing the distribution of colors within an image in a three-dimensional space. This involves separating the image into Red, Green, and Blue channels to study each color's contribution to the overall image. By comparing the 3D RGB cubes of the input (varnished) and output (restored) images, we can assess how the color distribution has changed due to the restoration process. This comparison can reveal whether the model is accurately restoring the original colors or introducing any color distortions. A well-restored image should have a color distribution closely matching the original unvarnished painting. By comparing the 3D RGB cubes of the input and output images, it can visually demonstrate the accuracy of the color restoration, as shown in Figures 10 and 11.

Table 1 Comparison with Other Works

No.	Method	PNSR (dB)	SSIM	Source
1	GAN – Artwork Restoration	28.90	0.53	Kumar & Gupta (2024)
2	VAE (Varian Autoencoder) – Old Photo Restoration	23.33	0.70	Wan et al. (2020)
3	Conditional GAN – Artwork Inpainting Restoration	N/A	0.795	Adhikary et al. (2021)
4	CycleGAN – Image Dehazing	15.54	0.66	Engin et al. (2018)
5	CNN – Based Inpainting	22.14	N/A	Zeng et al. (2020)
6	GAN – Mural Restoration	34.36	0.91	Li et al. (2021)
7	CycleGAN – Virtual Unvarnished Painting Restoration	36.95	0.998	*current work*

Table 1 compares various deep-learning approaches applied to color enhancement, image restoration, and artwork restoration. However, it is essential to note that this comparison does not utilize the same dataset across all models, which introduces certain limitations. Differences in datasets, preprocessing steps, and training parameters can significantly impact the comparability of the results. These variations may lead to discrepancies in performance metrics and outcomes, making it challenging to draw definitive conclusions about the relative effectiveness of each approach. Therefore, while the table offers valuable insights into the capabilities of different models, caution should be exercised when interpreting the results due to these inherent differences.

4. CONCLUSIONS

Comparing the CycleGAN model with traditional painting restoration methods and other state-of-the-art techniques helps to position the work within the broader field. This comparison highlights the strengths and weaknesses of our approach, providing valuable insights for future improvements. Traditional painting restoration methods rely heavily on manual techniques performed by skilled restorers. While these methods effectively preserve and restore artworks, they are often time-consuming and susceptible to human error. Key issues include maintaining consistency in color restoration and the potential for human error, which can lead to unintended alterations in the artwork. These challenges underscore the need for more efficient and reliable restoration techniques that complement or enhance traditional methods. Virtual restoration is a complementary tool to the physical restoration of artwork, offering insights into potential results without replacing the traditional restoration process. It gives restorers a preview of how a painting might look post-restoration, assisting in decision-making and planning.

In conclusion, our study effectively restored the color fidelity and nuances of historical paintings using CycleGAN. This model is a complementary tool to physical restoration, offering valuable

insights without replacing traditional methods. Our findings highlight CycleGAN's potential in digital art conservation and underscore the importance of ongoing research and interdisciplinary collaboration. When using CycleGAN for restoration, several limitations must be considered, particularly with paintings with additional layers or reconstructions obscuring the original artwork. New layers of paint can conceal original details or alter the original composition, making it challenging for the CycleGAN model to accurately predict details hidden beneath these overpainted layers. This research paves the way for further exploration in enhancing virtual restoration techniques, potentially integrating more advanced deep learning models and expanding datasets to include a wider range of artistic styles and conditions. By bridging the gap between traditional conservation practices and cutting-edge technology, CycleGAN provides art conservators with a valuable tool for informed decision-making and the preservation of cultural heritage.

Interdisciplinary collaboration is crucial in digital restoration research, as working with experts in computer vision, art history, and materials science can yield new insights and perspectives. These cross-disciplinary efforts can lead to more comprehensive restoration techniques that address the technical and artistic aspects of virtual painting restoration. Partnering with art historians and materials scientists enhances understanding of artworks' historical, aesthetic, and physical contexts, while collaboration with art conservation professionals ensures alignment with traditional practices. Overall, CycleGAN exemplifies how technology can transform art restoration, merging art and technology to preserve and revitalize paintings, thereby protecting cultural heritage for the future.

REFERENCES

Adhikary, A., Bhandari, N., Markou, E., & Sachan, S. (2021). ArtGAN: Artwork Restoration Using Generative Adversarial Networks. *2021 13th International Conference on Advanced Computational Intelligence (ICACI)*, 199–206. https://doi.org/10.1109/ICACI52617.2021.9435888

Engin, D., Genc, A., & Ekenel, H. K. (2018). Cycle-Dehaze: Enhanced CycleGAN for Single Image Dehazing. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 938–9388. https://doi.org/10.1109/CVPRW.2018.00127

Farajzadeh, N., & Hashemzadeh, M. (2021). A Deep Neural Network Based Framework for Restoring the Damaged Persian Pottery via Digital Inpainting. *Journal of Computational Science*, *56*, 101486. https://doi.org/10.1016/j.jocs.2021.101486

Kumar, P., & Gupta, V. (2024). Unpaired Image-to-Image Translation Based Artwork Restoration Using Generative Adversarial Networks. In *Smart Innovation, Systems and Technologies* (Vol. 372, pp. 581–591). https://doi.org/10.1007/978-981-99-6774-2_52

Li, J., Wang, H., Deng, Z., Pan, M., & Chen, H. (2021). Restoration of Non-Structural Damaged Murals in Shenzhen Bao'an Based on a Generator–Discriminator Network. *Heritage Science*, *9*(1), 6. https://doi.org/10.1186/s40494-020-00478-w

Maali Amiri, M., & Messinger, D. W. (2021). Virtual Cleaning of Works of Art Using Deep Convolutional Neural Networks. *Heritage Science*, *9*(1), 94. https://doi.org/10.1186/s40494-021-00567-4

Maali Amiri, M., & Messinger, D. W. (2023). Virtual Cleaning of Works of Art Using a Deep Generative Network: Spectral Reflectance Estimation. *Heritage Science*, *11*(1), 16. https://doi.org/10.1186/s40494-023-00859-x

Pietroni, E., & Ferdani, D. (2021). Virtual Restoration and Virtual Reconstruction in Cultural Heritage: Terminology, Methodologies, Visual Representation Techniques and Cognitive Models. *Information*, *12*(4), 167. https://doi.org/10.3390/info12040167

Sizyakin, R., Voronin, V. V., & Pizurica, A. (2022). Virtual Restoration of Paintings Based on Deep Learning. In W. Osten, D. Nikolaev, & J. Zhou (Eds.), *Fourteenth International Conference on Machine Vision (ICMV 2021)* (p. 60). SPIE. https://doi.org/10.1117/12.2624371

Wan, Z., Zhang, B., Chen, D., Zhang, P., Chen, D., Liao, J., & Wen, F. (2020). *Bringing Old Photos Back to Life*. http://arxiv.org/abs/2004.09484

Wang, H.-L., Han, P.-H., Chen, Y.-M., Chen, K.-W., Lin, X., Lee, M.-S., & Hung, Y.-P. (2018). Dunhuang Mural Restoration Using Deep Learning. *SIGGRAPH Asia 2018 Technical Briefs*, 1–4. https://doi.org/10.1145/3283254.3283263

Wang, J., Zhang, E., Cui, S., Wang, J., Zhang, Q., Fan, J., & Peng, J. (2023). GGD-GAN: Gradient-Guided Dual-Branch Adversarial Networks for Relic Sketch Generation. *Pattern Recognition*, *141*, 109586. https://doi.org/10.1016/j.patcog.2023.109586

Wu, Y., Wang, X., Li, Y., Zhang, H., Zhao, X., & Shan, Y. (2021). Towards Vivid and Diverse Image Colorization with Generative Color Prior. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 14357–14366. https://doi.org/10.1109/ICCV48922.2021.01411

Xiao, Y., Jiang, A., Liu, C., & Wang, M. (2019). Single Image Colorization via Modified Cyclegan. *2019 IEEE International Conference on Image Processing (ICIP)*, 3247–3251. https://doi.org/10.1109/ICIP.2019.8803677

Zeng, Y., Gong, Y., & Zeng, X. (2020). Controllable Digital Restoration of Ancient Paintings Using Convolutional Neural Network and Nearest Neighbor. *Pattern Recognition Letters*, *133*, 158–164. https://doi.org/10.1016/j.patrec.2020.02.033

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). *Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks*. https://doi.org/10.48550/arXiv.1703.10593

Zou, Z., Zhao, P., & Zhao, X. (2021). Virtual Restoration of the Colored Paintings on Weathered Beams in the Forbidden City Using Multiple Deep Learning Algorithms. *Advanced Engineering Informatics*, *50*, 101421. https://doi.org/10.1016/j.aei.2021.101421

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 100 – 109
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Comparison of KNN and Random Forest Algorithms on E-Commerce Service Chatbot

Fardan Zamakhsyari [(1)*], Bagas Adi Makayasa [(2)], R. Abudullah Hamami [(3)], Muhammad Tulus Akbar [(4)], Andi Cahyono [(5)], Amirullah [(6)], Muhammad Zida Hisyamuddin [(7)], Maria Ulfah Siregar [(8)]

[1] Department of Informatics Engineering, Sekolah Tinggi Teknologi Cahaya Surya, Kediri, Indonesia

[5] Department of Medical Informatics, Universitas Sains dan Teknologi Indonesia, Riau, Indonesia

[2,3,4,6,7,8] Department of Informatics, Universitas Islam Negeri Sunan Kalijaga, Yogyakarta, Indonesia

e-mail :
{masfardan99,bagas13am,alfalimbany,muhammadtulusa,andicahyono98,amrullmukminin,hisyam110699}@gmail.com, maria.siregar@uin-suka.ac.id.

* Corresponding author.

This article was submitted on 29 January 2024, revised on 16 December 2024, accepted on 16 December 2024, and published on 31 January 2025.

Abstract

Technology heavily influences our lives, with the expansion of e-commerce being an important outcome that demands attention. Given the prevalence of smartphones equipped with messaging apps and fast networks, people often utilize these platforms to communicate with sellers, offering a convenient way for sellers to engage efficiently with a diverse customer base. Recognizing this trend, there is a need for digital transformation of services to improve operational efficiency. Thus, this study aimed to compare the efficiency of classification algorithms in e-commerce service chatbots. The researcher used machine learning techniques with KNN and Random Forest algorithms in this case. To assess the feasibility of the application, the chatbot results will be tested using the confusion matrix method to assess accuracy. From this study, it was obtained that the KNN method and calculating word weight using TF-IDF produces an accuracy value of 71.4%, thus confirming its feasibility.

Keywords: Chatbot, E-Commerce, NLP, KNN, Random Forest

Abstrak

Teknologi sangat memengaruhi kehidupan kita, dengan perkembangan *e-commerce* menjadi salah satu hal penting yang patut diperhatikan. Dengan adanya *smartphone* yang dilengkapi aplikasi pesan dan jaringan cepat, orang sering memanfaatkan platform ini untuk berkomunikasi dengan penjual, memberikan cara yang nyaman bagi penjual untuk berinteraksi secara efisien dengan berbagai pelanggan. Menyadari tren ini, diperlukan transformasi digital layanan untuk meningkatkan efisiensi operasional. Oleh karena itu, penelitian ini bertujuan untuk membandingkan efisiensi algoritma klasifikasi dalam chatbot layanan *e-commerce*. Dalam penelitian ini, peneliti menggunakan teknik pembelajaran mesin dengan algoritma KNN dan Random Forest. Untuk menilai kelayakan aplikasi, hasil chatbot akan diuji menggunakan metode *confusion matrix* untuk menilai akurasi. Dari penelitian ini, diperoleh bahwa metode KNN dan perhitungan bobot kata menggunakan TF-IDF menghasilkan nilai akurasi sebesar 71,4%, sehingga mengonfirmasi kelayakannya.

Kata Kunci: Chatbot, *E-Commerce*, NLP, KNN, Random Forest

1. INTRODUCTION

The influence of technology permeates every facet of our lives, and the emergence of e-commerce stands out as a consequential outcome. In today's era, the widespread ownership of smartphones equipped with quick messaging and networking applications has become a norm. Individuals leverage these applications to interact with sellers, offering sellers a convenient means

to efficiently respond to a diverse customer base (Rakhra et al., 2021). Introducing a powerful tool in this context, chatbots are gaining prominence in the business sector for their potential to automate client service and streamline human efforts. These conversational agents are pivotal in bridging the gap between human-human and human-computer interaction, demonstrating their capability to comprehend the context and provide appropriate responses (Reddy Karri & Santhosh Kumar, 2020).

Customer service in e-commerce is crucial for assisting site users inquiring about the products and services offered there. However, there are restrictions on customer service, including working hours that are not every time and a lack of responsiveness in answering customers' questions and can impact the efficiency of e-commerce (Astuti & Fatchan, 2019). One may encounter problems when using traditional customer service operated by humans, such as time efficiency, long hold times, conventionality, and errors in the information provided can be easily solved using a chatbot (Wibowo et al., 2020).

Chatbot technology represents a specific application of Natural Language Processing (NLP). NLP, a field within science, delves into the study of communication between humans and computers through natural language (Rosyadi et al., 2020). Various technologies empowered by Artificial Intelligence (AI) include automation, Machine Learning (ML), Natural Language Processing (NLP), machine vision, expert systems, and robotics. Additionally, AI has profoundly influenced diverse facets of life, encompassing healthcare, education, business, finance, manufacturing, and law (Kumar & Ali, 2020).

Machine learning employs programmed algorithms to anticipate output values based on analyzed input data within a suitable range. In this study, a chatbot is developed using supervised techniques rooted in machine learning and natural language processing (Zhang et al., 2020). These supervised techniques involve a mathematical model comprising labeled inputs and anticipated outputs for predictive modeling. Commonly utilized algorithms include Nearest Neighbor, Decision tree, Support vector machines, Naïve Bayes, and linear regression (Tamizharasi et al., 2020). Subsequently, NLP explores computers' ability to comprehend and process human language to generate responses. Various methods are employed to grasp the words and intentions of a user within a given context, ranging from basic searching patterns of texts in user messages to more advanced artificial intelligence techniques applied to the language used by humans (Chandra et al., 2020).

The researchers see several studies that use several algorithms to apply machine learning classification models. In research, A. Wibisono explained that Based on the comparison of trial results, the Random Forest algorithm performance measure has better results than the Naïve Bayes, K-Nearest Neighbor, and Decision Tree algorithms with the k-fold cross-validation method. The Random Forest algorithm can provide an average accuracy result of 85.66% (Wibisono & Fahrurozi, 2019). Furthermore, according to K. Nugraha's research, the built chatbot system can work well and provide a maximum accuracy value of 53.48% (Nugraha & Sebastian, 2021). Tamizharasi, in his research, also shows the accuracy test value of the chatbot system with the accuracy results of KNN at 87.66% and Naïve Bayes at 81% (Tamizharasi et al., 2020).

The present study employs K-Nearest Neighbor (KNN) and Random Forest algorithms because of their proficiency in handling text-based classification problems, which are fundamental to creating an e-commerce service chatbot. The selection of KNN was based on its computational efficiency and simplicity, particularly when categorizing data with a basic distribution. KNN is a non-parametric method that can be used in chatbot systems with little data because it doesn't require assumptions about the data distribution (Jiang et al., 2018). However, when working with big or complicated datasets, KNN suffers from a huge rise in computation time.

Meanwhile, Random Forest offers benefits in terms of accuracy and robustness, particularly for more complicated datasets. Merging several decision trees lowers the possibility of overfitting, which frequently happens in models with just one decision tree (Hoekstra et al., 2022). Because

of this benefit, Random Forest is more reliable in making correct predictions, even when the data exhibits significant unpredictability.In this experimental study, researchers tried to use the K-Nearest Neighbors (KNN) and Random Forest algorithms in machine learning. These algorithms will be used in building an e-commerce service chatbot and tested to get stable results for analysis.

2. METHODS

2.1 Flow Research and Data Collection

The first stage is the literature review stage, where information on previous research related to the research to be used as a reference is sought. The needs analysis stage is conducted to formulate the needs used during the research. The system implementation stage involves integrating machine learning into the chatbot, which includes creating the machine learning, training the data set, and creating a chatting application for testing the chatbot (Intan, 2019). The evaluation stage follows, where the chatbot is checked for successful integration and operation, and the accuracy of the machine learning created in the chatbot is tested (Mohey, 2016). In addition, a model validation step was also conducted to ensure that the chatbot performance was as expected and that the model could be used effectively in practical situations.

Data collection aims to obtain interactions between the chatbot and the service users. The dataset is collected based on the number of questions in the e-commerce application related to products, orders, shipping, promotions, sellers, returns, and payment services. In this regard, the collection of questions is also based on the top questions that frequently occur and are repeatedly asked by users of the application before and after making transactions on the e-commerce application (Jadhav & Kalita, 2019).

2.2 Natural Language Processing (NLP)

Text processing using the natural language processing (NLP) process is used so that the chatbot can understand the language used by humans (Mathew et al., 2019). NLP tries to understand the language spoken by humans and classifies, analyzes, and responds to text input in the question column. Then, the chatbot tries to analyze the question and adjust it to the dataset. Python has a set of libraries that meet NLP needs. Chatbot data will be extracted from the NLP layer, and the extraction result will respond to the question given to the user (Rakhra et al., 2021).

One of the objectives of successful artificial intelligence is for computers to digest information to the point where they can carry out jobs that people can. Humans frequently exchange information through discussion, which is comparable to the task of asking and answering questions. The question-answering problem can be approached in several ways, including rule-based, extractive, and generative approaches (Rizqullah et al., 2023). The foundation of the rule-based approach is the use of predetermined patterns to elicit answers, based on research on rule-based question answering. Several investigations have used the extraction approach. The questions and answers in the extractive approach are based on a passage or context, and the responses are taken out of the context as answer spans (van Aken et al., 2019). NLP allows the chatbot to classify, analyze, and generate appropriate responses based on the text input. In this study, the following preprocessing techniques were implemented before the classification stage:

a) **Tokenization**: This technique was used to divide the text into smaller chunks, called tokens, like words or sentences. The phrase "What are the shipping options?" for example, is broken down into the tokens "What," "are," "the," "shipping," and "options. " (Baykara & Güngör, 2022).
b) **Stopword Removal**: Using a predetermined stopword list, common words like "and," "in," and "which," which don't significantly aid in the classification process, were eliminated. This phase enhances the emphasis on important words and helps reduce noise (Zhang et al., 2020).

c) **Stemming and Lemmatization**: Words were reduced to their most basic forms to lessen variance in text representation. For instance, "purchase" was shortened to "buy." Lemmatization uses language norms to guarantee more contextually accurate findings, while stemming offers a heuristic-based method (Yunanto et al., 2023).
d) **Text Representation**: Two methods were used to convert processed text data into numerical representations. The first method uses TF-IDF (Term Frequency-Inverse Document Frequency), which ranks more informative terms by weighing the importance of a document's words about the corpus. Next, the Bag of Words (BOW) is used. Without considering word order, this method displays text as a matrix of word frequencies. During the classification step, these numerical representations allow KNN and Random Forest algorithms to process data efficiently (Li & Zhang, 2023).
e) **Spelling Check**: A spelling correction module was incorporated into the preprocessing pipeline to fix typographical problems. This step guarantees that user searches that contain misspellings are nonetheless appropriately categorized (Wang et al., 2021).

Following preprocessing, a trained NLP model processed the text data to find patterns and forecast query categories. By using TF-IDF or BOW-based representations for this prediction, the chatbot could group inquiries into pre-established classes according to how closely they matched the labeled dataset. Based on these predictions, the chatbot gave pertinent answers (Dogan & Uysal, 2020). These preprocessing procedures greatly improve the chatbot's capacity to correctly read user input, which raises classification accuracy and improves user satisfaction.

2.3 KNN and Random Forest

KNN and the random forest are very simple and efficient machine-learning algorithms. KNN is very familiar in pattern recognition or design, and this algorithm is widely used where the input sample data and its classes are fixed. This algorithm works if new input data is obtained; it will be classified based on similarity to other data. For example, in asking about promotions that apply to the e-commerce application, data will be classified based on keywords that lead to stored datasets previously. The result is data that is relevant to or close to the question. The random forest is a machine learning algorithm commonly used to combine the output of several decision forests to achieve results to handle classification and regulation problems (Ahmad et al., 2022).

The K-Nearest Neighbor (KNN) algorithm is a machine learning algorithm used to classify new objects based on the distance between test data and training data (Rahardja et al., 2019). ja et al. This algorithm is a non-parametric technique because the classification of test data points depends on the nearest training data point without considering the data point parameters. The accuracy of this algorithm will be higher when it has large enough training data (Sukmandhani et al., 2023). The KNN algorithm uses the equation from the calculation of the Euclidean distance. In the process of designing the Random Forest algorithm model, the Random Forest algorithm model is built using 'n' Decision Tree trees by finding the best 'n' value, then the optimum results of the Random Forest algorithm, the CART algorithm as a criterion, and using the best splitter to get the best accuracy results in the case of coronary heart disease data (Singh et al., 2023).

2.4 Evaluation

Evaluating the Confusion Matrix is a method used to evaluate the classification results of a model. The Confusion Matrix is a table representing the number of correct and incorrect predictions made by a model. The Confusion Matrix consists of four parts: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) (Wijanarko & Afrianto, 2020).

a) True Positive (TP) is the number of correct predictions from the positive class. This indicates that the model has successfully and correctly classified the positive class.
b) False Positive (FP) is the number of incorrect predictions from the negative class. This indicates that the model has wrongly classified the negative class as positive.
c) True Negative (TN) is the number of correct predictions from the negative class. This indicates that the model has successfully classified the negative class correctly.

d) False Negative (FN) is the number of incorrect predictions from the positive class. This indicates that the model has wrongly classified the positive class as negative.

From these four parts, the success rate of a model in classifying data can be seen. The higher the values of TP and TN, the better the model is at classifying the data. On the other hand, the higher the values of FP and FN, the worse the model is in classifying the data. Evaluating the Confusion Matrix is very important because it can provide accurate information about the success rate of a model in classifying data. In addition, the Confusion Matrix can also be used to calculate metrics such as accuracy, sensitivity, and specificity, which can help evaluate a model's performance (Bird et al., 2023).

Accuracy is the success rate of a model in classifying data overall, calculated by adding TP and TN and dividing by the total number of data tested. Evaluating the Confusion Matrix is very useful in evaluating the performance of a model, especially for imbalanced data. Imbalanced data is data where the number of positive and negative classes is not balanced, which can cause the model to be too sensitive or specific. By using the Confusion Matrix, the accuracy, sensitivity, and specificity of a model can be known, allowing for improvement if necessary (Fachreza et al., 2023).

3. RESULTS AND DISCUSSION

3.1 Chatbot Concepts

In creating a chatbot, we apply several algorithms and frameworks that help optimize the use of chatbots. In this case, we create a scheme that can help process data so that it can be displayed in the form of a chatbot. We show it in Figure 1, which explains the chatbot's flow, starting from the user inputting data. The data will be processed using NLP, where the system will predict and provide answer recommendations based on the questions given by the user. After the data is processed, there is a spelling check so that the input words are correct, and then the data will be checked in the system database. If the question is in the database, the answer displayed is appropriate, but if the answer is wrong, it will provide a default answer from the system.

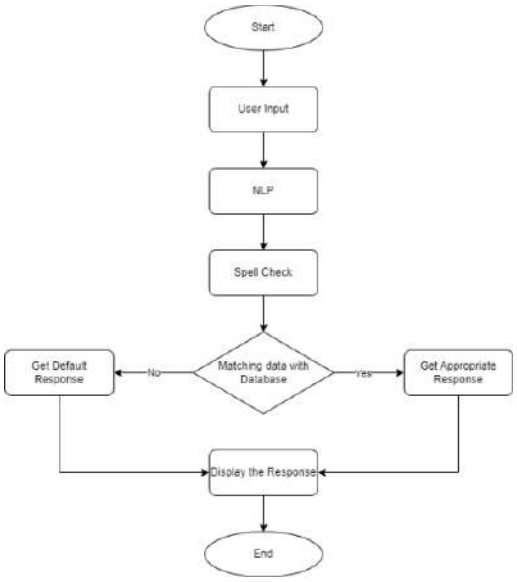


Figure 1 Chatbot Flowchart

3.2 Dataset Collection

The data collection stage is carried out by collecting Frequently Asked Question (FAQ) chat data from several e-commerce tools that can help customers. In addition, data is also collected through

observations related to e-commerce to several places and people who have a relationship in the buying and selling process. From this stage, 300 question data were collected, which will be used for the following process. The collected data is stored in a text file in JSON (JavaScript Object Notation) format as a knowledge database. The file structure consists of the 'knowledge' attribute, which contains classes resulting from the grouping of all the data that has been collected. Each class has a 'class' attribute to store the class name, 'patterns' to store a list of questions related to the class, and 'responses' contains a list of answers related to the class. The following dataset class distribution will be shown in Table 1.

Table 1 Database Class

Class	Description	Data
Description	Questions related to store description and information	50
Delivery	Questions related to store delivery service	50
Payment	Questions related to store types and method	50
Promo	Questions related to promos provided by the store	50
Return	Questions related to return service	50
Product Recommendation	Questions related to product recommendations from the store	50
	Total Data	**300**

3.3 Data Pre-processing

Preprocessing is a crucial step in Natural Language Processing (NLP) that guarantees machine learning models can understand the text. Tokenization, the first step in the preprocessing process in this study, divides the text into units of words called tokens. After that, frequent terms like "and," "in," and "which" that don't add anything to the classification are eliminated through the process of stopword removal. After that, words are reduced to their most basic form via stemming or lemmatization, for example, "purchase" becoming "buy". The TF-IDF and Bag of Words (BOW) approaches represent the text in numerical form; this step is crucial to enabling the KNN and Random Forest algorithms to process the text efficiently.

Following the completion of preprocessing, an NLP model is trained to identify patterns in the text and process the processed text. Based on the degree of similarity with the available dataset, these models predict the question's class using TF-IDF or BOW-based text representations. The chatbot responds appropriately based on these predictions. Additionally, the chatbot has a spelling check function that can handle typos in user input to increase accuracy.

3.4 Classification Model

The KNN approach for class construction creates a model based on a pre-specified number of k nearest neighbors. KNN uses the Euclidean distance to determine how far new data is from the remaining training data whenever it is received. When new data needs to be classified, this method computes the distance immediately rather than requiring explicit model training. In the meantime, several decision trees are created in Random Forest to build the model. Bootstrap sampling trains each tree with a random subset of the dataset. Each tree's construction features are likewise chosen at random. Different decision trees are produced by this method, which lessens overfitting and increases the stability of the model.

The KNN algorithm then uses the k nearest neighbors of the newly received data to make predictions for its prediction model. The final prediction will be determined by looking at the class that shows up the most among those neighbors. In contrast, every decision tree in the Random Forest makes a forecast on brand-new data. The majority votes on every decision tree to establish the final prediction. Compared to employing a single tree, this method guarantees stronger and more stable predictions.

3.5 Chatbot Display

This chatbot application is intended to help the efficiency and effectiveness of services in e-commerce, so the features and appearance are made according to needs. The results of the chatbot display will be shown in Figure 2. The chatbot display contains chats that have been sent and answers from the system occupying the right column, and the left contains the history of chats that have been sent. The display is made simple so that users can easily understand it. There is a settings menu at the bottom left. Besides that, users can also search for chat history on the lup icon on the top right.

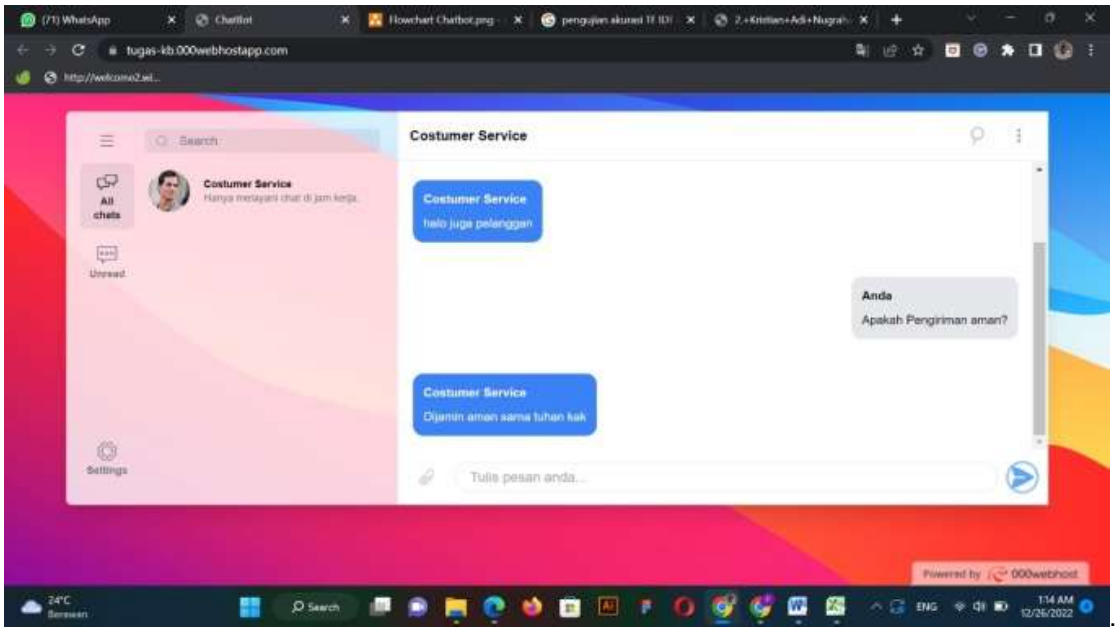


Figure 2 Chatbot Display

3.6 Accuracy Testing

Bot accuracy testing is carried out to determine the level of response accuracy the bot gives when users search using the bot application. This test is done by sending text messages directly to the bot. The test dataset used in this study is 70 pieces, and the following formula will obtain the accuracy value. Accuracy testing is conducted to assess how well the KNN and Random Forest models classify user queries. The models were tested using two text representations: TF-IDF and Bag of Words (BOW). In this test, there are several different test methods. The following results of the accuracy test will be displayed in Table 2.

$$Accuracy = \frac{Number\ of\ Correct\ Answers}{Total\ number\ of\ answers} \times 100\% \quad (1)$$

Table 2 Accuracy Testing

Testing	Correct Answer	Accuracy
KNN & TF-IDF	50	71,4%
KNN & BOW	37	52,7%
Random Forest & TF-IDF	43	61,4%
Random Forest & BOW	40	57,1%

Different accuracy results are obtained for each method, and the test is used based on the test results. Using the KNN method and calculating word weight using TF-IDF produces an accuracy

value of 71.4%. Then, the use of KNN and BOW methods yielded an accuracy of 52.8%. Using Random Forest and calculating word weights using TF-IDF produces an accuracy value of 61.4%. Moreover, finally, testing using Random forest and BOW has an accuracy value of 57.1%.

4. CONCLUSIONS

KNN and Random Forest algorithms to solve the e-commerce service chatbot problem. This research uses a dataset that we compiled and collected from various e-commerce with 300 questions. The measurements made by machine learning algorithms show the best accuracy and classification model. The highest accuracy value was obtained using the KNN method, and word weight calculation using TF-IDF resulted in an accuracy value of 71.4%, while testing using random forest and BOW yielded an accuracy value of 57.1%. This proves that the KNN Method and the calculation of word weights using TF-IDF are better at word processing.

Compared to previous studies, Tamizharasi et al. (2020) reported an accuracy of 87.66% using KNN with TF-IDF in the context of a medical chatbot. Although this study's accuracy is lower at 71.4%, the difference can be attributed to the domain-specific nature of datasets (e-commerce vs. medical) and the relatively smaller dataset used in this research. Furthermore, Nugraha & Sebastian (2021) achieved only 53.48% accuracy for a KNN-based chatbot, underscoring the effectiveness of the preprocessing and text representation methods applied in this study. These findings suggest that the KNN algorithm with TF-IDF is a promising approach for text-based classification in chatbot systems, particularly in domains with smaller datasets. However, the relatively lower performance of Random Forest in this study indicates that further exploration of hyperparameter tuning or feature engineering could improve its effectiveness.

This research contributes to the growing body of knowledge on e-commerce chatbot systems by demonstrating the effectiveness of KNN with TF-IDF for handling text classification tasks. For future research, the researcher recommends expanding the dataset to include more diverse and larger-scale e-commerce queries to improve generalization. The algorithms can use deep learning or neural networks to get maximum results. With these directions in mind, future research can build on the findings of this study to further improve the performance and applicability of chatbot systems in e-commerce.

REFERENCES

Ahmad, G. N., Ullah, S., Algethami, A., Fatima, H., & Akhter, S. Md. H. (2022). Comparative Study of Optimum Medical Diagnosis of Human Heart Disease Using Machine Learning Technique with and without Sequential Feature Selection. *IEEE Access*, *10*, 23808–23828. https://doi.org/10.1109/ACCESS.2022.3153047

Astuti, R. N., & Fatchan, M. (2019). Perancangan Aplikasi Teknologi Chatbot untuk Industri Komersial 4.0. *Prosiding Seminar Nasional Teknologi dan Sains (SNasTekS)*, *1*(1), 339–348. https://journal.unusida.ac.id/index.php/snts/article/view/103

Baykara, B., & Güngör, T. (2022). Abstractive Text Summarization and New Large-Scale Datasets for Agglutinative Languages Turkish and Hungarian. *Language Resources and Evaluation*, *56*(3), 973–1007. https://doi.org/10.1007/s10579-021-09568-y

Bird, J. J., Ekárt, A., & Faria, D. R. (2023). Chatbot Interaction with Artificial Intelligence: Human Data Augmentation with T5 and Language Transformer Ensemble for Text Classification. *Journal of Ambient Intelligence and Humanized Computing*, *14*(4), 3129–3144. https://doi.org/10.1007/s12652-021-03439-8

Chandra, A. Y., Kurniawan, D., & Musa, R. (2020). Perancangan Chatbot Menggunakan Dialogflow Natural Language Processing (Studi Kasus: Sistem Pemesanan pada Coffee Shop). *Jurnal Media Informatika Budidarma*, *4*(1), 208. https://doi.org/10.30865/mib.v4i1.1505

Dogan, T., & Uysal, A. K. (2020). A Novel Term Weighting Scheme for Text Classification: TF-MONO. *Journal of Informetrics*, *14*(4), 101076. https://doi.org/10.1016/j.joi.2020.101076

Fachreza, Moch. R. D., Suhartono, S., & Yaqin, M. A. (2023). Klasifikasi Sentimen Masyarakat terhadap Proses Pemindahan Ibu Kota Negara (IKN) Indonesia pada Media Sosial Twitter

Menggunakan Metode Naïve Bayes. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *8*(3), 243–251. https://doi.org/10.14421/jiska.2023.8.3.243-251

Hoekstra, O., Hurst, W., & Tummers, J. (2022). Healthcare Related Event Prediction from Textual Data with Machine Learning: A Systematic Literature Review. *Healthcare Analytics*, *2*, 100107. https://doi.org/10.1016/j.health.2022.100107

Intan, P. K. (2019). Comparison of Kernel Function on Support Vector Machine in Classification of Childbirth. *Jurnal Matematika "MANTIK,"* *5*(2), 90–99. https://doi.org/10.15642/mantik.2019.5.2.90-99

Jadhav, S. S., & Kalita, P. Ch. (2019). Design Thinking Approach in Planning E-commerce for Domestic Plumbing Services. *Proceedings of the 2019 International Conference on E-Business and E-Commerce Engineering*, 20–24. https://doi.org/10.1145/3385061.3385067

Jiang, M., Liang, Y., Feng, X., Fan, X., Pei, Z., Xue, Y., & Guan, R. (2018). Text Classification Based on Deep Belief Network and Softmax Regression. *Neural Computing and Applications*, *29*(1), 61–70. https://doi.org/10.1007/s00521-016-2401-x

Kumar, R., & Ali, M. M. (2020). A Review on Chatbot Design and Implementation Techniques. *International Research Journal of Engineering and Technology (IRJET)*, *7*(2), 2791–2800. https://www.irjet.net/archives/V7/i2/IRJET-V7I2592.pdf

Li, Q., & Zhang, Y. (2023). Improved Text Matching Model Based on BERT. *Frontiers in Computing and Intelligent Systems*, *2*(3), 40–43. https://doi.org/10.54097/fcis.v2i3.5209

Mathew, R. B., Varghese, S., Joy, S. E., & Alex, S. S. (2019). Chatbot for Disease Prediction and Treatment Recommendation Using Machine Learning. *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 851–856. https://doi.org/10.1109/ICOEI.2019.8862707

Mohey, D. (2016). Enhancement Bag-of-Words Model for Solving the Challenges of Sentiment Analysis. *International Journal of Advanced Computer Science and Applications*, *7*(1), 244–252. https://doi.org/10.14569/IJACSA.2016.070134

Nugraha, K. A., & Sebastian, D. (2021). Chatbot Layanan Akademik Menggunakan K-Nearest Neighbor. *Jurnal Sains dan Informatika*, *7*(1), 11–19. https://doi.org/10.34128/jsi.v7i1.285

Rahardja, C. A., Juardi, T., & Agung, H. (2019). Implementasi Algoritma K-Nearest Neighbor pada Website Rekomendasi Laptop. *Jurnal Buana Informatika*, *10*(1), 75–84. https://doi.org/10.24002/jbi.v10i1.1847

Rakhra, M., Gopinadh, G., Addepalli, N. S., Singh, G., Aliraja, S., Reddy, V. S. G., & Reddy, M. N. (2021). E-Commerce Assistance with a Smart Chatbot Using Artificial Intelligence. *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, 144–148. https://doi.org/10.1109/ICIEM51511.2021.9445316

Reddy Karri, S. P., & Santhosh Kumar, B. (2020). Deep Learning Techniques for Implementation of Chatbots. *2020 International Conference on Computer Communication and Informatics (ICCCI)*, *ICCCI*(2020), 1–5. https://doi.org/10.1109/ICCCI48352.2020.9104143

Rizqullah, M. R., Purwarianti, A., & Aji, A. F. (2023). QASiNa: Religious Domain Question Answering Using Sirah Nabawiyah. *2023 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*, 1–6. https://doi.org/10.1109/ICAICTA59291.2023.10390123

Rosyadi, H. E., Amrullah, F., Marcus, R. D., & Affandi, R. R. (2020). Rancang Bangun Chatbot Informasi Lowongan Pekerjaan Berbasis Whatsapp dengan Metode NLP (Natural Language Processing). *Briliant: Jurnal Riset dan Konseptual*, *5*(3), 619. https://doi.org/10.28926/briliant.v5i3.487

Singh, B., Olds, T., Brinsley, J., Dumuid, D., Virgara, R., Matricciani, L., Watson, A., Szeto, K., Eglitis, E., Miatke, A., Simpson, C. E. M., Vandelanotte, C., & Maher, C. (2023). Systematic Review and Meta-Analysis of the Effectiveness of Chatbots on Lifestyle Behaviours. *Npj Digital Medicine*, *6*(1), 118. https://doi.org/10.1038/s41746-023-00856-1

Sukmandhani, A. A., Lukas, Heryadi, Y., Suparta, W., & Wibowo, A. (2023). Classification Algorithm Analysis for Breast Cancer. *E3S Web of Conferences*, *388*, 02012. https://doi.org/10.1051/e3sconf/202338802012

Tamizharasi, B., Jenila Livingston, L. M., & Rajkumar, S. (2020). Building a Medical Chatbot using Support Vector Machine Learning Algorithm. *Journal of Physics: Conference Series*, *1716*(1), 012059. https://doi.org/10.1088/1742-6596/1716/1/012059

van Aken, B., Winter, B., Löser, A., & Gers, F. A. (2019). How does BERT Answer Questions? *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1823–1832. https://doi.org/10.1145/3357384.3358028

Wang, G., Cao, L., Zhao, H., Liu, Q., & Chen, E. (2021). Coupling Macro-Sector-Micro Financial Indicators for Learning Stock Representations with Less Uncertainty. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(5), 4418–4426. https://doi.org/10.1609/aaai.v35i5.16568

Wibisono, A. B., & Fahrurozi, A. (2019). Perbandingan Algoritma Klasifikasi dalam Pengklasifikasian Data Penyakit Jantung Koroner. *Jurnal Ilmiah Teknologi dan Rekayasa*, *24*(3), 161–170. https://doi.org/10.35760/tr.2019.v24i3.2393

Wibowo, B., Clarissa, H., & Suhartono, D. (2020). The Application of Chatbot for Customer Service in E-Commerce. *Engineering, MAthematics and Computer Science (EMACS) Journal*, *2*(3), 91–95. https://doi.org/10.21512/emacsjournal.v2i3.6531

Wijanarko, R., & Afrianto, I. (2020). Rancang Bangun Aplikasi Chatbot Media Informasi Parenting Pola Asuh Anak Menggunakan Line. *Matrix: Jurnal Manajemen Teknologi dan Informatika*, *10*(1), 1–10. https://doi.org/10.31940/matrix.v10i1.1805

Yunanto, R., Wibowo, E. P., & Rianto, R. (2023). A Bert Model to Detect Provocative Hoax. *Journal of Engineering Science and Technology*, *18*(5), 2281–2297. https://jestec.taylors.edu.my/Vol%2018%20Issue%205%20October%202023/18_5_03.pdf

Zhang, J., Zhang, J., Ma, S., Yang, J., & Gui, G. (2020). Chatbot Design Method Using Hybrid Word Vector Expression Model Based on Real Telemarketing Data. *KSII Transactions on Internet and Information Systems*, *14*(4), 1400–1418. https://doi.org/10.3837/tiis.2020.04.001

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 110 – 121
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Enhancing Abstractive Multi-Document Summarization with Bert2Bert Model for Indonesian Language

Aldi Fahluzi Muharam [(1)], Yana Adita Gerhana [(2)], Dian Sa'adillah Maylawati [(3)*], Muhammad Ali Ramdhani [(4)], Titik Khawa Abdul Rahman [(5)]
[1,2,3,4] Department of Informatics, Universitas Islam Negeri Sunan Gunung Djati, Bandung, Indonesia
[5] Department of Information and Communication Technology, Asia e University, Selangor, Malaysia
e-mail : 1207050008@student.uinsgd.ac.id,
{yanagerhana,diansm,m_ali_ramdhani}@uinsgd.ac.id, titik.khawa@aeu.edu.my.
* Corresponding author.
This article was submitted on 15 September 2024, revised on 19 Desember 2024, accepted on 28 Desember 2024, and published on 31 Januari 2025.

Abstract

This study investigates the effectiveness of the proposed Bert2Bert and Bert2Bert+Xtreme models in improving abstract multi-document summarization for Indonesians. This research uses the transformer model to develop the proposed Bert2Bert and Bert2Bert+Xtreme models. This research uses the Liputan6 data set which contains news data along with summary references for 10 years from October 2000 to October 2010 and is commonly used in many automatic text summarization research. The model evaluation results using ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore show that the proposed model has a slight improvement over previous research models, with Bert2Bert being better than Bert2Bert+Xtreme. Despite the challenges posed by limited reference summaries for Indonesian documents, content-based analysis using readability metrics, including FKGL, GFI, and Dwiyanto Djoko Pranowo, revealed that the summaries produced by Bert2Bert and Bert2Bert+Xtreme are at a moderate readability level, meaning they are suitable for mature readers and aligns with the news portal's target audience.

Keywords: Bert2Bert, Abstractive, Multi-document, Summarization, Transformer

Abstrak

Studi ini menyelidiki efektivitas model Bert2Bert dan Bert2Bert+Xtreme yang diusulkan dalam meningkatkan peringkasan multi dokumen yang abstrak untuk bahasa Indonesia. Penelitian ini menggunakan model transformer sebagai dasar untuk mengembangkan model Bert2Bert dan Bert2Bert+Xtreme yang diusulkan. Kumpulan data Liputan6 yang berisikan data berita beserta referensi ringkasannya dengan periode 10 tahun mulai dari Oktober 2000 sampai Oktober 2010 digunakan dalam penelitian ini. Hasil evaluasi model menggunakan ROUGE-1, ROUGE-2, ROUGE-L, dan BERTScore menunjukkan bahwa model yang diusulkan memiliki sedikit peningkatan terhadap model penelitian terdahulu dengan Bert2Bert lebih baik daripada Bert2Bert+Xtreme. Meskipun tantangan yang ditimbulkan oleh ringkasan referensi terbatas untuk dokumen-dokumen Indonesia, analisis berbasis konten menggunakan metrik keterbacaan termasuk FKGL, GFI, dan Dwiyanto Djoko Pranowo mengungkapkan bahwa ringkasan yang dihasilkan oleh Bert2Bert dan Bert2Bert+Xtreme berada pada tingkat keterbacaan sedang, yang berarti cocok untuk pembaca dewasa dan selaras dengan target audiens portal berita.

Kata Kunci: Bert2Bert, Abstraktif, Multi-dokumen, Peringkasan, Transformer

1. INTRODUCTION

Natural language processing applications are being developed using artificial intelligence, which allows computers to have natural language processing capabilities like humans (Alquliti & Binti, 2019), one of which is automatic text summarization. With the development of the information age, the need for content processing and summarization is increasing. Therefore, there is a need for a document summarization tool that can automatically summarize information from various sources. Document summarization tools aim to generate relevant and appropriate text summaries

for a given document or set (Jin & Wan, 2020). There are two types of document summarization: abstractive and extractive (Jin & Wan, 2020). Extractive summarization methods focus on identifying and compiling key sentences from the source text to form a summary. This approach benefits from simpler implementation and direct utilization of text from the original document (Kuyate et al., 2023).

Meanwhile, abstractive summarization generates new sentences, often requiring complex models to paraphrase and condense the original text's meaning, which can lead to summaries that are more fluent and less redundant (Dangol et al., 2023). Extractive summarization directly retrieves meaningful sentences the model has selected from the original document (Shinde et al., 2022). Summaries written by humans are generally abstractive. Abstractive summarization allows computers to generate text summaries by creating a new set of sentences that represent the information contained in the source with a different form of presentation from the original text (Jin & Wan, 2020; Li & Zhuge, 2021).

Every language has its uniqueness and characteristics, including Indonesian. Indonesian text summarization has seen significant advancements through the application of transformer models, addressing the challenges of processing a language spoken by nearly 200 million people but under-represented in NLP research (Devi & Suadaa, 2022). The existence of research on the development of Indonesian language data sets as an evaluation benchmark in the development of automatic text summarization tools opens opportunities for further research. IndoSum is a data set used as a new benchmark in Indonesian text summarization (Kurniawan & Louvan, 2018). Then, Liputan6 is used as a large-scale data collection tool for automatic text summarization (Koto, Rahimi, et al., 2020). Liputan6 was collected from data from an Indonesian language news portal, Liputan6.com, over ten years, so there were 215,827 text summary data documents (Koto, Rahimi, et al., 2020). In his research, a BERT-based single document summarization model was also developed using extractive and abstractive methods. MultilingualBERT and IndoBERT are used as pre-trained models in this model. Using IndoBERT produces good evaluation values for using the Liputan6 data set (Koto, Rahimi, et al., 2020). Then, there is also a comprehensive data set, Indonesia Language Evaluation Montage (IndoLEM), which includes seven NLP tasks and eight sub-data sets (Koto, Rahimi, et al., 2020). Other extractive text summarization models for the Indonesian language, such as the fine-tuning of Sentence Transformers (SBERT), have demonstrated improved performance in generating document summaries or snippets, particularly using Indonesian thesis documents to construct a new dataset for this task (Abka et al., 2022).

Based on the development of research on automatic text summarization for Indonesian, it was found that abstractive multi-document summarization in Indonesian research has not been carried out using a Transformer-based model, and it is a great opportunity to explore. This is due to the limited data sets that provide multi-document summary data (Jin & Wan, 2020; J. Zhang et al., 2018). In cases other than Indonesian, there is research on fine-tuning using the BERT Sentence Embedding Model on extractive summarization multi-document data sets (Lamsiyah et al., 2023). Then, there is also research by fine-tuning the Transformers model, which has been trained for single-document summarization of multi-document data sets by adjusting the Encoder and Decoder structure (Shen et al., 2023). Other research also adapts the Transformer model trained for single-document summarization to perform multi-document summarization tasks using the Decoding Controller (Jin & Wan, 2020). All three studies used pre-trained models that were fine-tuned using multi-document data. From these researches, it emerged that the single-document summarization model has the potential to perform multi-document summarization tasks. Therefore, this study contributes to using a model trained with a single document summarization data set in an Indonesian multi-document summarization task with the Transformers model.

2. METHODS

2.1 Datasets

This research uses the Liputan6 data set in the training and evaluation process because the data set is 11 times larger and more abstract than the IndoSum data set (Koto, Rahimi, et al., 2020;

Lucky & Suhartono, 2021). There are 193,883 training data and 10,972 data for development and testing, respectively (Koto, Rahimi, et al., 2020). Each document has data for article text, extractive summary text, and abstractive summary text. The development and testing data have a more significant percentage of novel n-grams than the training data. Apart from the Canonical variant, there is an Xtreme variant in development and testing data, namely a data set that only contains more than 90% novel 4-grams. Hence, the data is more abstractive and produces a smaller data set. In the Liputan6 training data set, the number of words in article documents ranges from the smallest, 31 words, to the largest, 6,570 words, with an average of 195.74 words, a mode of 121 words, and a median of 163 words. Meanwhile, in reference documents, the abstractive summaries range from the smallest being 11 words to the largest being 80 words, with an average of 27.08 words, a mode of 27 words, and a median of 27 words.

The Canonical variant development data set on article text documents has the smallest 64 words and the largest, 1,567 words, with an average of 190.1 words, a mode of 141 words, and a median of 166 words. In the abstract summary document, the number of words ranges from 9 to 36, with an average of 21.88 words, a mode of 23, and a median of 22. For the Xtreme variant development data set, the smallest number of article text document words is 66 words, and the largest is 1567 words, with an average of 196.55 words, a mode of 141 words, and a median of 168 words. The number of words in abstractive summary text documents consists of the smallest 11 words, the largest 36 words, the average 21.16 words, the mode 22 words, and the median 21 words.

Meanwhile, in the Canonical variant test data set, the number of words in the article text document ranges from the smallest 62 words to the largest, 3,064 words, with an average of 181.53 words, a mode of 150 words, and a median of 158 words. The abstractive summary text ranges from 12 words to 35 words with an average of 23.18 words, a mode of 24 words, and a median of 23 words. In the Xtreme variant, the number of words in the article text document ranges from 63 to 3064 words with an average of 194.81 words, a mode of 132 words, and a median of 161 words. For abstractive summary text, the range is from 12 words to 33 words, with an average of 22.62 words, a mode of 22 words, and a median of 23 words.

2.2 Bert2Bert Model

Transformer models thrive on automatic text summarization as NLP tasks. Transformer-based models are one type of model that can be used with the sequence-to-sequence learning (seq2seq) method. Using seq2seq-based models shows promising results on single document summarization tasks (Jin & Wan, 2020; Koto, Rahimi, et al., 2020; J. Zhang et al., 2018). The seq2seq model is a model that can be trained using varying input dimensions and with varying output dimensions as well. IndoBERT is applied as an encoder and decoder transformer with random or non-pretrained parameters to develop a single document summarization model. This research implements IndoBERT as an Encoder and Decoder using the Huggingface framework. BERT is a model designed to train in-depth two-way representations of unlabeled text by considering context from left and right at all layers (Devlin et al., 2018). The BERT model architecture represents the Encoder layer in a Transformer. The Encoder layer in the Transformer consists of a multi-head self-attention layer and a position-wise fully connected feed-forward network layer (Vaswani et al., 2017). Meanwhile, the Decoder layer in the Transformer is like the Encoder layer but modified by masking and adding a third layer, the multi-head cross-attention layer, which receives input from the Encoder output. Masking the Decoder layer, the multi-head self-attention layer that was originally bidirectional become unidirectional.

Figure 1 shows the architecture of the transformer model where the blue layer shows the parameter weights initiated randomly. In contrast, the red layer shows the parameter weights using the weights from the pre-trained BERT model. In the process of implementing BERT at the Encoder layer, there are no problems because the structure is the same. However, several adjustments are made to the model layer structure in the BERT implementation at the Decoder layer. In this process, a cross-attention layer is added between the self-attention and feed-forward layers (Rothe et al., 2020). The parameter weights in this additional layer are initiated randomly.

Changes to the self-attention layer from bi-directional to unidirectional were also carried out without changing the parameter weights of that layer. Then, the LM (Language Model) Head layer was added as the final layer to define the conditional probability distribution of the output sequence. The parameter weights of this layer use the word embedding weights in the BERT Embedding layer.

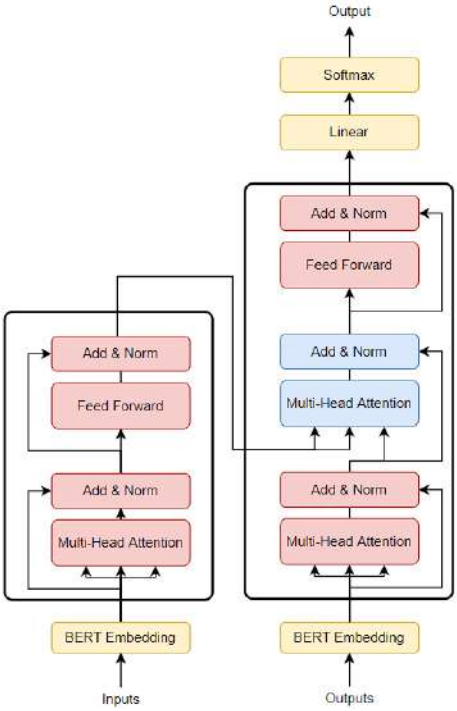


Figure 1 Transformer Model with IndoBERT Implementation

The tokenizer from IndoBERT is used by setting the maximum Encoder token length to 512, the maximum Decoder token length to 256, and by adjusting the decoder_start_token_id with bos_token_id from the tokenizer. In the model settings also, adjustments are made to eos_token_id with eos_token_id from the tokenizer, pad_token_id with pad_token_id on the tokenizer, and vocab_size with encoder_vocab_size. Then, in the generative configuration, the maximum length is set to 80, the minimum length is 10, num_beams is 10, length_penalty is 2, no_repeat_n_gram_size is 3, and early_stopping is set to true. Adam optimization is used with a learning rate of 5×10-5 and lr_scheduler_type linear.

Two training methods are used in model development. In the first scenario, the model is trained with 8 epochs using 193,883 training data and a batch size of 18. Then, in the second scenario, the first model trained in the first scenario is trained again using the abstractive Xtreme variant development data set with 5 epochs and a batch size of 10. This data set has more than 90% novel 4-grams in each document, so it is assumed to produce a model that works more abstractive.

2.3 Evaluation Methods

The evaluation of model capabilities is divided into two categories: co-selection-based analysis and content-based analysis (Maylawati et al., 2024). Co-selection-based analysis compares the summary results against the reference summary. Meanwhile, the content-based analysis assesses the readability of summary results through sentence linkages without requiring a reference summary. The co-selection-based evaluation process was carried out using ROUGE and BERTScore using single-document data from the Liputan6 data test. BERTScore is a text generation evaluation metric using context-based token representation to measure the similarity

of text to its reference (T. Zhang et al., 2019). The evaluation process is carried out using the F1 value from BERTScore. Then ROUGE or Recall-Oriented Understudy for Gisting Evaluation is a tool for measuring summarization results by comparing other ideal human summaries (Lin, 2004). This research uses three F1 values, namely ROUGE-1, ROUGE-2, and ROUGE-L, to measure the quality of the summarization results. ROUGE-1 and ROUGE-2 measure using N-grams that intersect each other between the summarization results and the reference with length N of 1 and 2 (Lin, 2004; Maylawati et al., 2024). The ROUGE-L measurement is based on the Longest Common Subsequence (LCS) or the longest substitution (Lin, 2004; Maylawati et al., 2024). This concept considers sentence structure in identifying similarities to consider words that are not sequential but still have the same or similar meaning in the context of the sentence.

Content-based evaluation in multi-document summarization results uses Flesch-Kincaid Grade Level (FKGL), Gunning Fog Index (GFI), and Dwiyanto Djoko Pranowo metric values. FKGL is a metric for measuring a text's difficulty level in understanding based on the length of words and sentences (Solnyshkina et al., 2017). Like FKGL, GFI is a metric used to measure a text's difficulty level in reading (Świeczkowski & Kułacz, 2021). Among the FKGL and FGIeveloped in English, Dwiyanto Djoko Pranowo is the creator of special measuring tools in Indonesian texts (Biddinika et al., 2016).There are thirteen indicators to assess readability in the Dwiyanto metric. The thirteen indicators are the average number of paragraphs, the average number of sentences in each paragraph, sentence length, percentage of continuation sentences, percentage of compound sentences, percentage of number of polysemic sentences, percentage of passive sentences, percentage of foreign words, percentage of abstract words, percentage of terms, percentage conjunctions, and percentage of loanwords (Pranowo, 2011). The readability measure is obtained by adding up all these indicators. The range 13.0 to 21.7 is defined as easy, the range 21.8 to 30.5 is interpreted as moderate, and the range 30.6 to 39.0 is interpreted as difficult.

3. RESULTS AND DISCUSSION

3.1 Result of Fine-Tuning Model

Previous research using IndoBERT for automatic text summarization was carried out for single documents (Koto, Rahimi, et al., 2020). Therefore, to determine the performance of the proposed model, namely Bert2Bert, the experiment was carried out using a single document with the same dataset as previous research, namely Liputan6 data. This model can accept input with a maximum length of 512 tokens. With that, documents whose length exceeds 512 after tokenization will have the excess removed. The model was then tested using Liputan6 test data, and the evaluation results are shown in Table 1. The evaluation was carried out using ROUGE and BERTScore.

Table 1 Evaluation result using ROUGE and BERTScore

Model	Canonical Test Set				Extreme Test Set			
	R1	R2	RL	BS	R1	R2	RL	BS
BERTABS (Koto, Lau, et al., 2020)	40.94	23.01	37.89	77.90	34.59	15.10	31.19	75.84
BERTEXTABS (Koto, Lau, et al., 2020)	41.08	22.85	38.01	77.93	34.84	15.03	31.40	75.99
Bert2Bert	41.13	22.85	34.65	73.30	34.86	15.29	27.89	72.54
Bert2Bert+Xtreme	39.21	20.04	32.51	72.66	34.45	14.56	27.37	72.10

In Table 1, the proposed model (Bert2Bert) has a higher R1 value in Canonical data testing, namely 41.13. It has the largest R1 and R2 values in the Xtreme data test, namely 34.86 and 15.29. Although the ROUGE and BERTScore results on Bert2Bert are like IndoBert, overall, Bert2Bert performs well like other models. This is indicated by the ROUGE and BERTScore values, which are not significantly different in the Canonical and Xtreme tests. Compared with the Bert2Bert+Xtreme model, another proposed model, the Bert2Bert model, performs better. This is shown in the ROUGE and BERTScore results, which are both better than those of

Bert2Bert+Xtreme. Table 2 shows examples of abstractive summary results produced by Bert2Bert and Bert2Bert+Xtreme.

Table 2 The Example of an Abstractive Summary with Bert2Bert

Reference Summary	Bert2Bert	Bert2Bert+Xtreme
Kendati Bank Sentral AS menurunkan suku bunganya, namun BI dinilai masih akan menemui masa sulit. Suku bunga Bank Sentral AS akan diturunkan menjadi empat persen. (Even though the US Central Bank has lowered its interest rate, BI is still considered to be facing difficult times. The US Central Bank interest rate will be reduced to four percent.)	menurut pengamat ekonomi didiek j. rachbini, bank indonesia akan menghadapi situasi sulit bila bi terus menurunkan tingkat suku bunga yang dimiliki.	bank indonesia dinilai masih akan menghadapi situasi sulit di tanah air. bahkan, tingkat suku bunga the fed akan diturunkan menjadi empat persen.
Kapolda Riau baru Brigjen Pol. Johny Yodjana bertekad memberantas pelaku penyelundupan kayu di Riau. Ia berjanji akan menindak tegas pelaku tanpa pandang bulu. (The new Riau Regional Police Chief, Brigadier General Pol. Johny Yodjana, is determined to eradicate wood smuggling perpetrators in Riau. He promised to take firm action against the perpetrators without discrimination.)	polda riau bertekad memberantas para penyelundup kayu di riau. kapolda riau brigjen pol. johny yodjana berjanji tak akan pandang bulu.	kapolda riau brigjen pol. johny yodjana melantik kapolda baru di riau. selain itu, polri juga akan memberantas penipuan dana reboisasi dan iuran hasil hutan.

3.2 Result of Abstractive Multi-Document Summarization for Indonesian Language

The Bert2Bert and Bert2Bert+Extreme models have been proven to generate automatic abstractive summaries for single documents. Furthermore, this research applies the Bert2Bert and Bert2Bert+Extreme models for multi-document abstractive summarization in Indonesian using Transformers, which has never been done in previous research. Previous research conducts multi-document abstractive summarization without Transformers (Severina & Khodra, 2019). Then, most of the last research only focused on Indonesian abstractive summarization for single documents (Devianti & Khodra, 2019; Dewi & Widiastuti, 2022; Laksana et al., 2022; Sugiri et al., 2022; Wijayanti et al., 2021). Several previous multi-document studies also did not produce abstractive summarization but extractive summarization (D. Gunawan et al., 2019; Y. H. B. Gunawan & Khodra, 2021; Widjanarko et al., 2018). Table 3 and 4 shows the example of abstractive summarization results for Indonesian multi-documents with 2 Canonicals, 2 Xtremes, 3 Canonocals, and 3 Xtremes, where the bold means selected by Bert2Bert while underlined is selected by Bert2Bert+Xtreme.

3.3 Readability Evaluation

The limited number of summary references for multiple Indonesian documents makes conducting co-selection-based analysis evaluations such as ROUGE and BERTScore impossible. Therefore, the abstractive summary results of multiple Indonesian documents were evaluated using content-based analysis such as the readability metrics FKGL, GFI, and Dwiyanto Djoko Pranowo. Readability of summary results is a big challenge in automatic text summarization research, which is currently the focus of researchers (Maylawati, 2019; Verma et al., 2019; Verma & Om, 2019). Most automatic text summarization research involves humans evaluating the summary results, but more is needed on readability. As one of the contributions to this research, this section presents the results of multi-document abstractive summaries with Bert2Bert and Bert2Bert+Extreme along with the results of readability evaluation using FKGL, GFI, and Dwiyanto Djoko Pranowo metrics.

Figures 2 (a) and (b) show the FKGL and GFI evaluation results of Bert2Bert and Bert2Bert+Xtreme with Canonical and Xtreme data. Based on the FKGL evaluation results, Bert2Bert and Bert2Bert+Xtreme have a readability level of more than 18 for FKGL, which means

the resulting text is difficult to read or understand. The FKGL value indicates a text's readability based on the reader's age grade. For FKGL scores, 0-6 are categorized as easy, 6-12 as average, 12-18 as poor reading skills, and above 18 are considered difficult to read (Maylawati et al., 2024; Scott, 2024; Solnyshkina et al., 2017). The optimal value of GFI for the text that is categorized as readable is in the range of 7 to 8 (Maylawati et al., 2024; Scott, 2025; Świeczkowski & Kułacz, 2021). However, the average GFI value is 9-10, which means that readability is still acceptable because, according to GFI, texts that are very difficult to read have a value of more than 12. However, overall, the FKGL and GFI results cannot be categorized as hard to read to the adult age category because the data source is news portals whose readers are adults. So, the FKGL and GFI evaluation results can be accepted by news readers who are mostly adults.

Table 3 The Example of an Abstractive Summary with Bert2Bert

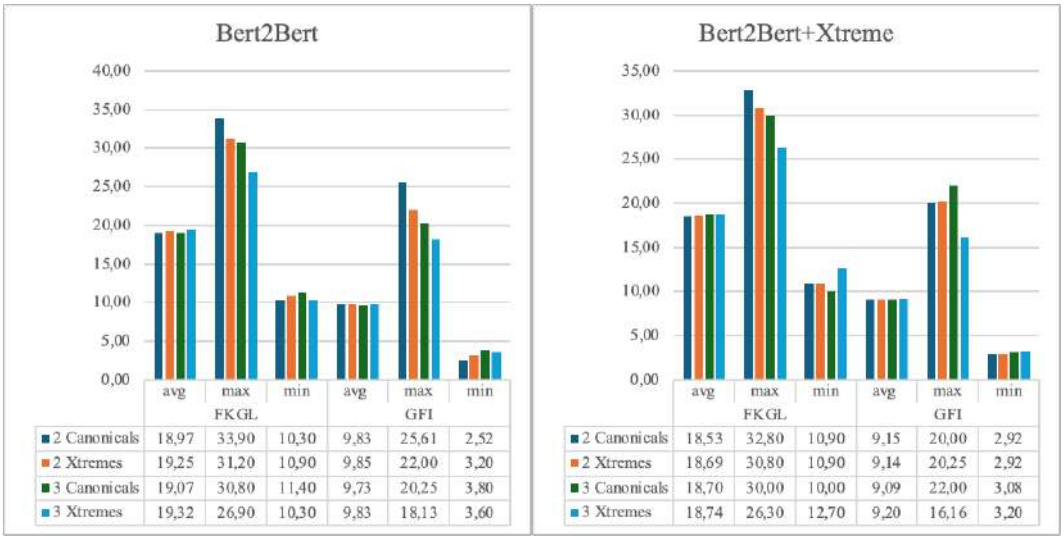
Type	Articles	Bert2Bert Summary	Bert2Bert+Extreme Summary
2 Canonicals	[Doc 1] Liputan6.com, Jakarta: **Kepolisian Daerah Riau bertekad memberantas pelaku penyelundupan kayu yang kerap terjadi di Riau.** Selain itu, Polda setempat juga akan memberangus menipulasi dana reboisasi dan iuran hasil hutan. Demikian ditegaskan Kepala Polda Riau Brigadir Jenderal Polisi Johny Yodjana, seusai dilantik menjadi Kapolda Riau oleh Kepala Polri Jenderal Polisi Suroyo Bimantoro, di Jakarta, baru-baru ini. Menurut Johny, pelaku tindak kriminal yang kerap menjarah kayu di Riau akan ditindak tegas. "Saya tak akan pandang bulu," janji Johny. (ICH/Edi Priyono dan Andi Azril). [Doc 2] Liputan6.com, Jakarta: **Bank Indonesia dinilai masih akan menghadapi situasi sulit kendati Bank Sentral Amerika Serikat (The FED) terus menurunkan tingkat suku bunga yang dimiliki.** Penilaian itu dikemukakan pengamat ekonomi Didiek J. Rachbini, di Jakarta, baru-baru ini. Menurut perhitungan Didiek, dalam tahun ini, The FED telah lima kali menurunkan nilai suku bunga yang mereka miliki. Bahkan, Didiek memperkirakan, tingkat suku bunga The FED akan diturunkan hingga menjadi empat persen. Dengan keadaan itu, tambah Didiek, di atas kertas dapat dimanfaatkan BI untuk meningkatkan suku bunga BI sebagai upaya mempertahankan nilai tukar rupiah. Namun demikian, Didiek pesimistis, hal itu akan tercapai mengingat kondisi bangsa masih carut marut. "Jika keadaan terus seperti ini, tak tertutup kemungkinan, BI akan tetap memberlakukan nilai suku bunga tinggi," ujar Didiek. ICH/Fahmi Ihsan dan Donny Indradi).	polda riau bertekad memberantas para penyelundup kayu di riau. dalam tahun ini, the fed telah menurunkan tingkat suku bunga yang dimiliki the fed.	kapolda riau melantik kapolda baru menggantikan gubernur no. h. thobrak. didiek j. rachbini optimistis, bi akan tetap mempertahankan nilai suku bunga the fed.
2 Xtremes	[Doc 1] Liputan6.com, Jakarta: Romadhani alias Roban, **penjahat kelas kakap, tewas tertembak sesaat sebelum beraksi di kawasan Pasar Induk Kramatjati, Jakarta Timur,** Senin (31/12). Pria berusia 30 tahun tewas setelah peluru polisi bersarang di dadanya. Kepala Unit Reserse dan Intelijen Kepolisian Resor Jaktim Inspektur Satu Polisi Sudiono mengatakan, Roban terpaksa ditembak karena melawan ketika hendak ditangkap. (ZAQ/Nurul Amin dan Gatot Setiawan). [Doc 2] Liputan6.com, Jakarta: Menurut perhitungan Didiek, dalam tahun ini, **The FED telah lima kali menurunkan nilai suku bunga yang mereka miliki**. Bahkan, Didiek memperkirakan, tingkat suku bunga The FED akan diturunkan hingga menjadi empat persen. Dengan keadaan itu, tambah Didiek, di atas kertas dapat dimanfaatkan BI untuk meningkatkan suku bunga BI sebagai upaya mempertahankan nilai tukar rupiah. Namun demikian, Didiek pesimistis, hal itu akan tercapai mengingat kondisi bangsa masih carut marut. "Jika keadaan terus seperti ini, tak tertutup kemungkinan, BI akan tetap memberlakukan nilai suku bunga tinggi," ujar Didiek. (ICH/Fahmi Ihsan dan Donny Indradi).	seorang penjahat kelas kakap tewas ditembak di kawasan pasar induk kramatjati, jaktim. dalam tahun ini, the fed telah lima kali menurunkan suku bunga yang dimiliki the fed.	seorang penjahat kelas kakap tewas ditembak polisi saat beraksi di pasar kramatjati, jaktim. didiek j. rachbini pesimistis, bi akan mampu menurunkan suku bunga bank indonesia.

Table 4 The Example of an Abstractive Summary with Bert2Bert (Continued)

Type	Articles	Bert2Bert Summary	Bert2Bert+Extreme Summary
3 Canonicals	[Doc 1] Liputan6.com, Jakarta: Operasi Sadar Jaya yang dilancarkan Selasa (15/5) malam, sekitar pukul 23. 00 WIB, mengejutkan pengunjung Diskotik Millenium, yang berlokasi di Jalan Gajah Mada, Jakarta Pusat. Sebanyak 200 petugas gabungan dari Kepolisian Resor Metro Jakarta Pusat dan kesatuan Brigade Mobil Polda Metro Jaya menggeledah seluruh pengunjuk diskotik yang tengah asyik berdansa. Dari operasi tersebut, polisi menangkap 32 **pengunjung diskotik yang tertangkap basah membawa 66 butir pil ekstasi.** 14 orang di antara mereka adalah wanita muda. Para pengunjung yang tertangkap tampak pasrah saat dibawa ke kantor polisi. Sementara itu, kaca mobil patroli Polres Metro Jakpus yang digunakan untuk razia, terlihat pecah karena dilempar batu. Kaca mobil patroli pecah saat polisi menggeledah Diskotik Millenium. (COK/Christiyanto dan Johni Akbar). [Doc 2] Liputan6.com, Tangerang: **Empat warga negara asing terdakwa penyelundup heroin disidangkan di Pengadilan Negeri Tangerang**, Banten, Selasa (15/5). Keempat orang itu adalah Samuel Uwuchukwu Okoye dari Nigeria, Ozias Sibanda dari Zimbabwe, Hansen Antony Nwaolisa, dan Okwudili Ayotanze dari Liberia. Keempat tersangka diancam hukuman mati. (ICH/Roy Akhmad dan Agung Nugroho). [Doc 3] Liputan6.com, Jakarta: Tunggakan Kredit Usaha Tani (KUT) di Bank Rakyat Indonesia mencapai Rp 2, 4 triliun. Akibatnya, penyaluran Kredit Ketahanan Pangan (KKP) sebagai pengganti KUT untuk petani juga ikut terhambat. Demikian dikemukakan Direktur Utama BRI Rudjito, Rabu (16/5). Rudjito mengutip catatan pemerintah bahwa tunggakan KUT untuk musim tanam tahun 2000 mencapai Rp 3, 2 triliun. Sekitar 75 persen di antaranya KUT yang disalurkan BRI. (YYT/Olivia Rosalia dan Donni Indradi).	sebanyak 32 pengunjung diskotik millenium, jakpus, ditangkap karena kedapatan membawa 66 butir pil ekstasi. empat terdakwa penyelundup heroin disidangkan di pn tangerang.	diskotik millenium yang terletak di jalan gajah mada, jakarta pusat, dirazia ratusan polisi. empat tersangka penyelundup heroin disidangkan di pengadilan negeri tangerang.
3 Xtremes	[Doc 1] Liputan6.com, Jakarta: Romadhani alias Roban, **penjahat kelas kakap, tewas tertembak sesaat sebelum beraksi di kawasan Pasar Induk Kramatjati.** Jakarta Timur. Senin (31/12). Pria berusia 30 tahun tewas setelah peluru polisi bersarang di dadanya. Kepala Unit Reserse dan Intelijen Kepolisian Resor Jaktim Inspektur Satu Polisi Sudiono mengatakan, Roban terpaksa ditembak karena melawan ketika hendak ditangkap. (ZAQ/Nurul Amin dan Gatot Setiawan). [Doc 2] Liputan6.com, Jakarta: Operasi Sadar Jaya yang dilancarkan Selasa (15/5) malam, sekitar pukul 23. 00 WIB, mengejutkan pengunjung Diskotik Millenium, yang berlokasi di Jalan Gajah Mada, Jakarta Pusat. Sebanyak 200 petugas gabungan dari Kepolisian Resor Metro Jakarta Pusat dan kesatuan Brigade Mobil Polda Metro Jaya menggeledah seluruh pengunjuk diskotik yang tengah asyik berdansa. (COK/Christiyanto dan Johni Akbar). [Doc 3] Liputan6.com, Jakarta: **Tunggakan Kredit Usaha Tani (KUT) di Bank Rakyat Indonesia mencapai Rp 2, 4 triliun. Akibatnya, penyaluran Kredit Ketahanan Pangan (KKP) sebagai pengganti KUT untuk petani juga ikut terhambat.** Demikian dikemukakan Direktur Utama BRI Rudjito, Rabu (16/5). Rudjito mengutip catatan pemerintah bahwa tunggakan KUT untuk musim tanam tahun 2000 mencapai Rp 3, 2 triliun. (YYT/Olivia Rosalia dan Donni Indradi).	seorang penjahat yang kerap beraksi di pasar kramatjati, jakarta timur, tewas ditembak polisi. tunggakan kut di bri sebesar rp 2, 4 triliun membuat penyaluran kredit terhambat.	seorang penjahat kelas kakap tewas ditembak polisi di kawasan pasar induk kramatjati, jakarta timur. tunggakan kredit usaha tani untuk petani juga menghambat penyaluran kredit ketahanan pangan.

FKGL and GFI are readability measurement metrics used for English. However, several linguistic studies have adapted itnd is suitable for Indonesians (Mursyadah, 2021; Sari & Herri, 2020; Utami et al., 2021). A readability metric is used specifically for the Indonesian language, namely the Dwiyanto Djoko Pranowo metric. Therefore, this study's summary results of Bert2Bert and Bert2Bert+Xtreme also used Dwitanto Djoko Pranowo to measure readability. The evaluation results show that the summary of Bert2Bert and Bert2Bert+Xtreme, shown in Table 5, are in the

moderate readability range, namely 20.59 and 22.17. This indicates that the summary results of Indonesian multi-documents using Bert2Bert and Bert2Bert+Xtreme have good readability and are still understandable.



(a) (b)

Figure 2 FKGL and GFI Result: (a) Bert2Bert, (b) Bert2Bert+Xtreme

Table 5 Dwitanto's Evaluation of Bert2Bert and Bert2Bert+Extreme

Indicators	Bert2Bert				Bert2Bert+Xtreme			
	2 Canonicals	2 Xtremes	3 Canonicals	3 Xtremes	2 Canonicals	2 Xtremes	3 Canonicals	3 Xtremes
	Average of Dwiyanto's Score				Average of Dwiyanto's Score			
Paragraph	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Sentence Count	1.95	1.95	1.95	1.95	2.07	2.09	2.08	2.11
Sentence Length	12.80	12.76	12.40	12.19	11.54	11.41	11.37	10.97
Extension	0.74	0.75	0.75	0.76	0.74	0.74	0.75	0.75
Compound	0.76	0.76	0.76	0.77	0.75	0.75	0.75	0.77
Polysemy	0.76	0.76	0.76	0.77	0.76	0.76	0.76	0.77
Passive Sentence	0.63	0.63	0.63	0.64	0.62	0.63	0.63	0.64
Unfamiliar Word	0.01	0.01	0.02	0.02	0.01	0.02	0.02	0.02
Abstract Word	0.72	0.73	0.72	0.73	0.72	0.72	0.72	0.73
Terms	0.77	0.77	0.77	0.78	0.76	0.76	0.77	0.78
Conjunctions	0.74	0.74	0.75	0.75	0.73	0.73	0.73	0.74
Loan	0.70	0.70	0.70	0.70	0.69	0.69	0.69	0.70
Phrase	0.60	0.60	0.60	0.61	0.59	0.59	0.59	0.60
Dwiyanto's Total Score	22.17	22.17	21.82	21.66	20.99	20.87	20.85	20.59

4. CONCLUSIONS

The research findings highlight the performance of the proposed model, Bert2Bert, showcasing its efficacy in abstractive multi-document summarization tasks. While the ROUGE and BERTScore metrics show similarity between Bert2Bert and IndoBERT, Bert2Bert stands out with superior performance, especially compared to the Bert2Bert+Xtreme model. Despite the limitations in conducting co-selection-based analysis evaluations like ROUGE and BERTScore due to the lack of summary references for multiple Indonesian documents, this research adopts content-based analysis using readability metrics such as FKGL, GFI, and Dwiyanto Djoko

Pranowo. The results show that Bert2Bert and Bert2Bert+Xtreme produce summaries with an appropriate level of readability for adult readers, as expected from the target audience of the news portal. Overall, the moderate readability range of summary results suggests that the Bert2Bert and Bert2Bert+Xtreme models offer summaries that are understandable and accessible to their intended audience, aligning with the nature of news content targeted at adult readers. Future research could explore other transformer models for abstract summaries of several documents in Indonesian, such as GPT2GPT, BERT2GPT, or GPT2BERT. Further research could also contribute to preparing higher quality datasets for automatic text abstraction summarization of multiple documents to provide a more comprehensive evaluation using co-selection-based metrics and further improve model performance. By overcoming these challenges, future research can contribute to advancing automatic document summarization technology and be applied to real-world cases in Indonesia.

ACKNOWLEDGEMENT

We want to thank the various parties who supported this research, especially the Department of Informatics, UIN Sunan Gunung Djati Bandung, which funded the publication of this research.

REFERENCES

Abka, A. F., Azizah, K., & Jatmiko, W. (2022). Transformer-based Cross-Lingual Summarization Using Multilingual Word Embeddings for English - Bahasa Indonesia. *International Journal of Advanced Computer Science and Applications*, *13*(12). https://doi.org/10.14569/IJACSA.2022.0131276

Alquliti, W. H., & Binti, N. (2019). Convolutional Neural Network Based for Automatic Text Summarization. *International Journal of Advanced Computer Science and Applications*, *10*(4), 200–211. https://doi.org/10.14569/IJACSA.2019.0100424

Biddinika, M. K., Lestari, R. P., Indrawan, B., Yoshikawa, K., Tokimatsu, K., & Takahashi, F. (2016). Measuring the Readability of Indonesian Biomass Websites: The Ease of Understanding Biomass Energy Information on Websites in the Indonesian Language. *Renewable and Sustainable Energy Reviews*, *59*, 1349–1357. https://doi.org/10.1016/j.rser.2016.01.078

Dangol, R., Adhikari, P., Dahal, P., & Sharma, H. (2023). Short Updates-Machine Learning Based News Summarizer. *Journal of Advanced College of Engineering and Management*, *8*(2), 15–25. https://doi.org/10.3126/jacem.v8i2.55939

Devi, K. U. S., & Suadaa, L. H. (2022). Extractive Text Summarization for Snippet Generation on Indonesian Search Engine Using Sentence Transformers. *2022 International Conference on Data Science and Its Applications (ICoDSA)*, 181–186. https://doi.org/10.1109/ICoDSA55874.2022.9862886

Devianti, R. S., & Khodra, M. L. (2019). Abstractive Summarization Using Genetic Semantic Graph for Indonesian News Articles. *2019 International Conference on Advanced Informatics: Concepts, Theory, and Applications, ICAICTA 2019*, 1–6. https://doi.org/10.1109/ICAICTA.2019.8904361

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. https://doi.org/10.48550/arXiv.1810.04805

Dewi, K. E., & Widiastuti, N. I. (2022). The Design of Automatic Summarization of Indonesian Texts Using a Hybrid Approach. *Jurnal Teknologi Informasi dan Pendidikan*, *15*(1), 37–43. https://doi.org/10.24036/jtip.v15i1.451

Gunawan, D., Harahap, S. H., & Fadillah Rahmat, R. (2019). Multi-document Summarization by using TextRank and Maximal Marginal Relevance for Text in Bahasa Indonesia. *2019 International Conference on ICT for Smart Society (ICISS)*, *7*, 1–5. https://doi.org/10.1109/ICISS48059.2019.8969785

Gunawan, Y. H. B., & Khodra, M. L. (2021). *Multi-document Summarization Using Semantic Role Labeling and Semantic Graph for Indonesian News Article*. https://doi.org/10.48550/arXiv.2103.03736

Jin, H., & Wan, X. (2020). Abstractive Multi-document Summarization via Joint Learning with Single-Document Summarization. *Findings of the Association for Computational Linguistics: EMNLP 2020*, *2020*, 2545–2554. https://doi.org/10.18653/v1/2020.findings-emnlp.231

Koto, F., Lau, J. H., & Baldwin, T. (2020). *Liputan6: A Large-scale Indonesian Dataset for Text Summarization*. http://arxiv.org/abs/2011.00679

Koto, F., Rahimi, A., Lau, J. H., & Baldwin, T. (2020). IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP. *Proceedings of the 28th International Conference on Computational Linguistics*, 757–770. https://doi.org/10.18653/v1/2020.coling-main.66

Kurniawan, K., & Louvan, S. (2018). Indosum: A New Benchmark Dataset for Indonesian Text Summarization. *2018 International Conference on Asian Language Processing (IALP)*, 215–220. https://doi.org/10.1109/IALP.2018.8629109

Kuyate, S., Jadhav, O., & Jadhav, P. (2023). AI Text Summarization System. *International Journal for Research in Applied Science and Engineering Technology*, *11*(5), 916–919. https://doi.org/10.22214/ijraset.2023.51481

Laksana, M. D. B., Karyawati, A. E., Putri, L. A. A. R., Santiyasa, I. W., Sanjaya ER, N. A., & Kadnyanan, I. G. A. G. A. (2022). Text Summarization terhadap Berita Bahasa Indonesia Menggunakan Dual Encoding. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, *11*(2), 339. https://doi.org/10.24843/JLK.2022.v11.i02.p13

Lamsiyah, S., Mahdaouy, A. El, Ouatik, S. E. A., & Espinasse, B. (2023). Unsupervised Extractive Multi-document Summarization Method Based on Transfer Learning from BERT Multi-task Fine-Tuning. *Journal of Information Science*, *49*(1), 164–182. https://doi.org/10.1177/0165551521990616

Li, W., & Zhuge, H. (2021). Abstractive Multi-Document Summarization Based on Semantic Link Network. *IEEE Transactions on Knowledge and Data Engineering*, *33*(1), 43–54. https://doi.org/10.1109/TKDE.2019.2922957

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*, 74–81. https://aclanthology.org/W04-1013/

Lucky, H., & Suhartono, D. (2021). Investigation of Pre-Trained Bidirectional Encoder Representations from Transformers Checkpoints for Indonesian Abstractive Text Summarization. *Journal of Information and Communication Technology*, *21*(1), 71–94. https://doi.org/10.32890/jict2022.21.1.4

Maylawati, D. S. (2019). Sequential Pattern Mining and Deep Learning to Enhance Readability of Indonesian Text Summarization. *International Journal of Advanced Trends in Computer Science and Engineering*, *8*(6), 3147–3159. https://doi.org/10.30534/ijatcse/2019/78862019

Maylawati, D. S., Kumar, Y. J., Kasmin, F., & Ramdhani, M. A. (2024). Deep Sequential Pattern Mining for Readability Enhancement of Indonesian Summarization. *International Journal of Electrical and Computer Engineering (IJECE)*, *14*(1), 782. https://doi.org/10.11591/ijece.v14i1.pp782-795

Mursyadah, U. (2021). Tingkat Keterbacaan Buku Sekolah Elektronik (BSE) Pelajaran Biologi Kelas X SMA/MA. *TEACHING: Jurnal Inovasi Keguruan dan Ilmu Pendidikan*, *1*(4), 298–304. https://doi.org/10.51878/teaching.v1i4.774

Pranowo, D. D. (2011). *Alat Ukur Keterbacaan Teks Berbahasa Indonesia*. Universitas Negeri Yogyakarta.

Rothe, S., Narayan, S., & Severyn, A. (2020). Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. *Transactions of the Association for Computational Linguistics*, *8*, 264–280. https://doi.org/10.1162/tacl_a_00313

Sari, M. P., & Herri, H. (2020). Analisa Konten Serta Tingkat Keterbacaan Pernyataan Misi dan Pengaruhnya terhadap Kinerja Perbankan Indonesia. *Menara Ilmu: Jurnal Penelitian dan Kajian Ilmiah*, *14*(1), 96–106. https://doi.org/10.31869/mi.v14i1.2003

Scott, B. (2024). *Learn How to Use the Flesch-Kincaid Grade Level Formula*. ReadabilityFormulas.Com. https://readabilityformulas.com/learn-how-to-use-the-flesch-kincaid-grade-level/

Scott, B. (2025). *The Gunning Fog Index (or FOG) Readability Formula*. ReadabilityFormulas.Com. https://readabilityformulas.com/the-gunnings-fog-index-or-fog-readability-formula/

Severina, V., & Khodra, M. L. (2019). Multidocument Abstractive Summarization Using Abstract Meaning Representation for Indonesian Language. *2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, 1–6. https://doi.org/10.1109/ICAICTA.2019.8904449

Shen, C., Cheng, L., Nguyen, X.-P., You, Y., & Bing, L. (2023). *A Hierarchical Encoding-Decoding Scheme for Abstractive Multi-document Summarization*. https://doi.org/10.48550/arXiv.2305.08503

Shinde, K., Roy, T., & Ghosal, T. (2022). An Extractive-Abstractive Approach for Multi-document Summarization of Scientific Articles for Literature Review. *Proceedings of the Third Workshop on Scholarly Document Processing*, 204–209. https://aclanthology.org/2022.sdp-1.25/

Solnyshkina, M. I., Zamaletdinov, R. R., Gorodetskaya, L. A., & Gabitov, A. I. (2017). Evaluating Text Complexity and Flesch-Kincaid Grade Level. *Journal of Social Studies Education Research*, *8*(3), 238–248. http://www.jsser.org/index.php/jsser/article/view/225

Sugiri, Eko Prasojo, R., & Alfa Krisnadhi, A. (2022). Controllable Abstractive Summarization Using Multilingual Pretrained Language Model. *2022 10th International Conference on Information and Communication Technology (ICoICT)*, 228–233. https://doi.org/10.1109/ICoICT55009.2022.9914846

Świeczkowski, D., & Kułacz, S. (2021). The Use of the Gunning Fog Index to Evaluate the Readability of Polish and English Drug Leaflets in the Context of Health Literacy Challenges in Medical Linguistics: An Exploratory Study. *Cardiology Journal*, *28*(4), 627–631. https://doi.org/10.5603/CJ.a2020.0142

Utami, S. D., Dewi, I. N., & Efendi, I. (2021). Tingkat Keterbacaan Bahan Ajar Flexible Learning Berbasis Kolaboratif Saintifik. *Bioscientist : Jurnal Ilmiah Biologi*, *9*(2), 577. https://doi.org/10.33394/bioscientist.v9i2.4246

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. https://doi.org/10.48550/arXiv.1706.03762

Verma, P., & Om, H. (2019). A Novel Approach for Text Summarization Using Optimal Combination of Sentence Scoring Methods. *Sādhanā*, *44*(5), 110. https://doi.org/10.1007/s12046-019-1082-4

Verma, P., Pal, S., & Om, H. (2019). A Comparative Analysis on Hindi and English Extractive Text Summarization. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *18*(3), 1–39. https://doi.org/10.1145/3308754

Widjanarko, A., Kusumaningrum, R., & Surarso, B. (2018). Multi Document Summarization for the Indonesian Language Based on Latent Dirichlet Allocation and Significance Sentence. *2018 International Conference on Information and Communications Technology (ICOIACT)*, 520–524. https://doi.org/10.1109/ICOIACT.2018.8350668

Wijayanti, R., Khodra, M. L., & Widyantoro, D. H. (2021). Indonesian Abstractive Summarization Using Pre-trained Model. *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, 79–84. https://doi.org/10.1109/EIConCIT50028.2021.9431880

Zhang, J., Tan, J., & Wan, X. (2018). *Towards a Neural Network Approach to Abstractive Multi-Document Summarization*. https://doi.org/10.48550/arXiv.1804.09010

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). BERTScore: Evaluating Text Generation with BERT. *8th International Conference on Learning Representations, ICLR 2020*. https://doi.org/10.48550/arXiv.1904.09675

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 122 – 138
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Android Malware Threats: A Strengthened Reverse Engineering Approach to Forensic Analysis

Ridho Surya Kusuma [(1)*], M Dirga Purnomo Putra [(2)]
Department of Informatics, Universitas Siber Muhammadiyah, Yogyakarta, Indonesia
e-mail : {ridhosuryakusuma, dirga20220100042}@sibermu.ac.id
* Corresponding author.
This article was submitted on 29 August 2024, revised on 30 January 2025, accepted on 30 January 2025, and published on 31 January 2025.

Abstract

The widespread adoption of Android devices has rendered them a primary target for malware attacks, resulting in substantial financial losses and significant breaches of user privacy. Malware can exploit system vulnerabilities to execute unauthorized premium SMS transactions, exfiltrate sensitive data, and install additional malicious applications. Conventional detection methodologies, such as static and dynamic analysis, often prove inadequate in identifying deeply embedded malicious behaviors. This study introduces a systematic reverse engineering framework for analysing suspicious Android applications. In contrast to traditional approaches, the proposed methodology consists of six distinct stages: Initialization, decompilation, static analysis, code reversing, behavioral analysis, and reporting. This structured process facilitates a comprehensive examination of an application's internal mechanisms, enabling the identification of concealed malware functionalities. The findings of this study demonstrate that the proposed method attains an overall effectiveness of 84.3%, surpassing conventional static and dynamic analysis techniques. Furthermore, this research generates a detailed list of files containing specific malware indicators, thereby enhancing future malware detection and prevention systems. These results underscore the efficacy of reverse engineering as a critical tool for understanding and mitigating sophisticated Android malware threats.

Keywords: Malware Android, Reverse Engineering, Android Security, Digital Forensic, Cybersecurity

Abstrak

Adopsi perangkat Android yang meluas telah menjadikannya target utama serangan malware, yang mengakibatkan kerugian finansial yang besar dan pelanggaran signifikan terhadap privasi pengguna. Malware dapat mengeksploitasi kerentanan sistem untuk melakukan transaksi SMS premium yang tidak sah, mengeksfiltrasi data sensitif, dan memasang aplikasi berbahaya tambahan. Metodologi deteksi konvensional, seperti analisis statis dan dinamis, sering kali terbukti tidak memadai dalam mengidentifikasi perilaku berbahaya yang tertanam dalam. Penelitian ini memperkenalkan kerangka kerja reverse engineering yang sistematis untuk analisis aplikasi Android yang mencurigakan. Berbeda dengan pendekatan tradisional, metodologi yang diusulkan terdiri dari enam tahap yang berbeda: Inisialisasi, dekompilasi, analisis statis, pembalikan kode, analisis perilaku, dan pelaporan. Proses terstruktur ini memfasilitasi pemeriksaan komprehensif terhadap mekanisme internal aplikasi, memungkinkan identifikasi fungsi malware yang tersembunyi. Temuan penelitian ini menunjukkan bahwa metode yang diusulkan mencapai efektivitas keseluruhan sebesar 84,3%, melampaui teknik analisis statis dan dinamis konvensional. Selain itu, penelitian ini menghasilkan daftar file terperinci yang berisi indikator malware tertentu, sehingga berkontribusi pada peningkatan sistem pendeteksian dan pencegahan malware di masa mendatang. Hasil ini menggarisbawahi keampuhan reverse engineering sebagai alat penting untuk memahami dan memitigasi ancaman malware Android yang canggih.

Kata Kunci: Malware Android, *Reverse Engineering*, *Android Security*, *Digital Forensic*, *Cybersecurity*

1. INTRODUCTION

In cybersecurity, the prevalence of Android malware attacks represents a significant and escalating threat to the security of users, organizations, and the integrity of digital ecosystems. While previous studies have extensively highlighted (Manzil & Manohar Naik, 2023; Qamar et al., 2019), limited attention has been given to integrating advanced detection techniques with scalable forensic methods to address these challenges effectively. This research aims to bridge that gap by proposing a unified approach that examines the intricacies of Android malware behavior and develops enhanced methodologies for malware detection and forensic investigation (Umar et al., 2021b).

Unlike earlier works that primarily focused on specific malware categories, such as banking malware or SMS malware, this study offers a comprehensive approach combining static and dynamic analysis methods to detect disguised and stealthy malware. Previous studies, such as those by Joseph Raymond and Jeberson Joseph Raymond & Jeberson Retna Raj (2023) and Liu et al. (2023), have explored the application of machine learning and deep learning in isolation. In contrast, this research integrates these approaches with novel forensic techniques like Just-in-Time Memory Forensics (JIT-MF), enabling effective real-time evidence collection during malware incidents (Bellizzi et al., 2022).

A key distinction of this study lies in its dual-layered methodology. While static analysis has proven effective for identifying vulnerabilities and data leaks in Android applications (Almomani et al., 2022), dynamic analysis is employed here to address runtime behaviors and the limitations of static approaches. This combination enhances malware detection capabilities and establishes a robust framework for defense mechanisms, particularly for Android devices integrated with IoT applications—a domain identified as particularly vulnerable (Ashawa & Morris, 2021).

Researchers have explored innovative approaches such as deep learning, machine learning, and dynamic analysis technologies to address the challenges posed by Android malware. For instance, Alkahtani & Aldhyani (2022) demonstrated that deep learning models significantly enhance malware detection accuracy, particularly with large-scale datasets. Similarly, Ye et al. (2022) highlighted how machine learning algorithms effectively identify previously unseen malware variants, reducing the risk of zero-day attacks. However, while these methodologies represent critical advancements, they often fail to address real-time detection and evidence collection challenges—an issue this study directly tackles.

Furthermore, the emergence of disguised Android malware employing advanced evasion techniques has underscored the need for hybrid detection approaches. Elsersy et al. (2022) proposed frameworks integrating static and dynamic analysis to overcome individual method limitations. Building on this, Bellizzi et al. (2022) introduced JIT-MF as an innovative solution for collecting volatile memory evidence during active malware incidents. This research further expands upon these contributions by integrating JIT-MF with scalable detection frameworks, ensuring timely and efficient forensic investigations.

Despite significant advancements, a critical gap persists in integrating existing techniques into a unified framework that improves detection capabilities and ensures scalability and efficiency in forensic practices. This study addresses this gap by proposing a cohesive methodology that enhances both malware detection and forensic investigation processes, thereby strengthening the security of Android devices and the sensitive data they store. Rather than comparing the efficacy of detection methods—such as dynamic analysis or machine learning—this research focuses on identifying behavioral patterns and deriving forensic insights through reverse engineering techniques.

2. METHODS

The research method in this study uses a reverse engineering approach. The approach is a fundamental technique in Android malware forensics, which can extract digital evidence from

malicious apps (Qiu et al., 2019). This process is crucial for uncovering how malware works, identifying its behaviour, and extracting valuable insights that can help investigate and mitigatendroid malware attacks (Joseph Raymond & Jeberson Retna Raj, 2023). Reverse engineering is essential to overcome the challenges posed by sophisticated Android malware, including obfuscation techniques and encryption technologies designed to evade detection and Analysis (Ye et al., 2022) and (Urooj et al., 2022).

The flowchart below illustrates the sequential steps involved in the reverse engineering methodology for analyzing Android malware in digital forensics, as shown in Figure 1. Figure 1 describes the stages of the reverse engineering methodology in Android malware forensics, which consists of six main steps:

a) The initialiszation, which is the first step in acquiring and sampling Android malware, usually in the form of APK files (Lubuva et al., 2019).
b) Decompilation is decompiling the APK file to convert the binary code into a human-readable format.
c) Static Analysis is an analyst who examines decompiled code to determine malicious behaviour, such as data exfiltration, privilege escalation, and communication with command and control servers (Bhandari & Jusas, 2020b).
d) Code Reversing is an analysis that delves deeper into decompiled code through code reversal techniques to trace execution flow, identify encryption methods, and uncover obfuscation techniques used by malware to avoid detection (Mastino et al., 2022).
e) Behavioural Analysis is a comprehensive behavioural analysis that outlines the actions performed by the malware, including file modification, network communication, and system interaction (Serketzis et al., 2019).
f) Reporting is the process of documenting the results and pattern recognition of a detailed report covering the malware's characteristics, behaviour, and potential impact (Bhandari & Jusas, 2020a). This report is significant evidence for further investigation and mitigation efforts (Bartliff et al., 2020; Kusuma, 2023).

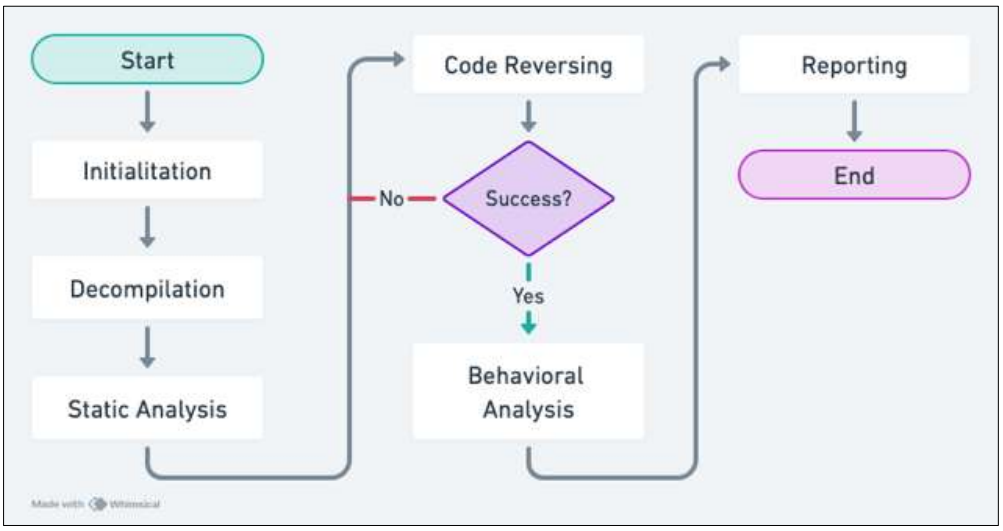


Figure 1 Flowchart of Reverse Engineering Method

A structured forensic methodology can effectively dissect Android malware, thus improving the ability to combat evolving cyber threats (Umar et al., 2021a). The following are details of the use of software in this research based on web applications, as shown in Table 1. Table 1 shows three web application-based tools are used to support the analysis and investigation of Android malware. The first tool is Decompiler (www.decompiler.com), which decompiles APK files into source code for further analysis. With this capability, Decompiler makes it easier to understand the structure and behavior of the analyzed application, especially to identify potentially suspicious activity.

Table 1 Web Application-Based Tools

Software	Description	Main Function	URL
Decompiler	A web-based tool used to analyze and decompile executable files.	Decompile Android application (APK) files into source code for further analysis.	www.decompiler.com
Metadefender	A cloud-based multi-scan platform that uses various antivirus engines for analysis.	Detects malware, vulnerabilities, and potential threats in APK files or others.	www.metadefender.com
Koodous	A community-based service that combines APK analytics with crowdsourced intelligence.	Detects malware in Android apps and provides deeper analysis of APK security.	www.koodous.com

The second tool is Metadefender (metadefender.com), a cloud-based multi-scan platform that uses various antivirus engines to detect malware and vulnerabilities in files. With its cloud-based approach, Metadefender enables more comprehensive detection as it utilizes technologies from various security providers, increasing the accuracy of the analysis. The third tool is Koodous (koodous.com), a community-based platform that offers security analysis for APK files. Koodous combines technical analysis with crowdsourced data from the security community, providing additional insights into malware and other security threats. The platform is particularly useful in detecting malicious apps not identified by other tools, thanks to active community contributions.

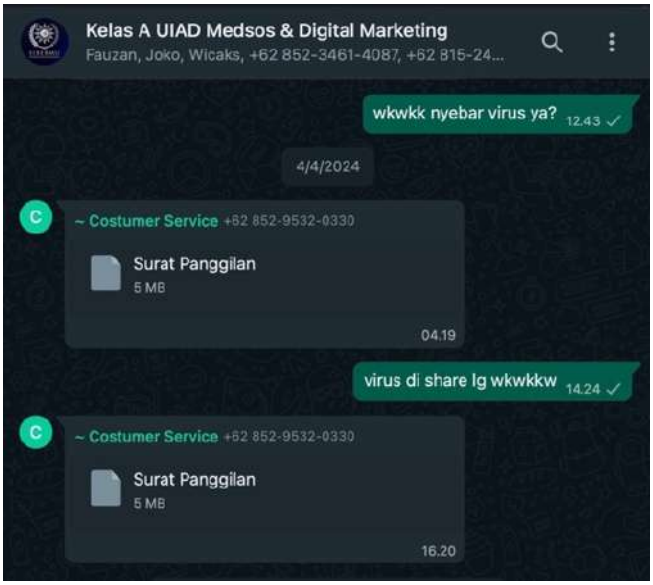


Figure 2 Deployment of WhatsApp.apk

These three tools are used synergistically to ensure that Android malware analysis is thorough, whether through decompilation, antivirus-based detection or community-based insights. The combination of these tools provides a strong foundation for investigating and mitigating security threats on the Android platform. The case study in this research is based on a real case in the course WhatsApp group and the use of datasets from Kaggle sourced from URLs: https://www.kaggle.com/datasets/saurabhshahane/android-malware-dataset to measure the effectiveness of this research method. The chronology begins when one of the member numbers in the group with the pseudonym 'Customer Service' sends a file, as shown in Figure 2. Based on Figure 2, the appearance of a file sent via WhatsApp with a name that looks like an official

document, namely 'Surat Panggilan.apk'. This naming can trick other group members into opening the file and running it directly. This file has a '.APK' format, indicating an unknown and suspicious Android program.

3. RESULTS AND DISCUSSION

This research conducts digital forensics with reverse engineering Android APK files originating from WhatsApp groups. The file named 'Surat Panggilan.APK' looks suspicious, so it needs further examination. The following are the findings of this research.

3.1 Initialization

The first stage of the digital forensics process involves computer-acquiring suspicious files. The file identification process uses Metadefender WebApps to measure the level of danger, file size, requested permissions, and suspicious code snippets. The results of the file examination can be seen in Figure 3.

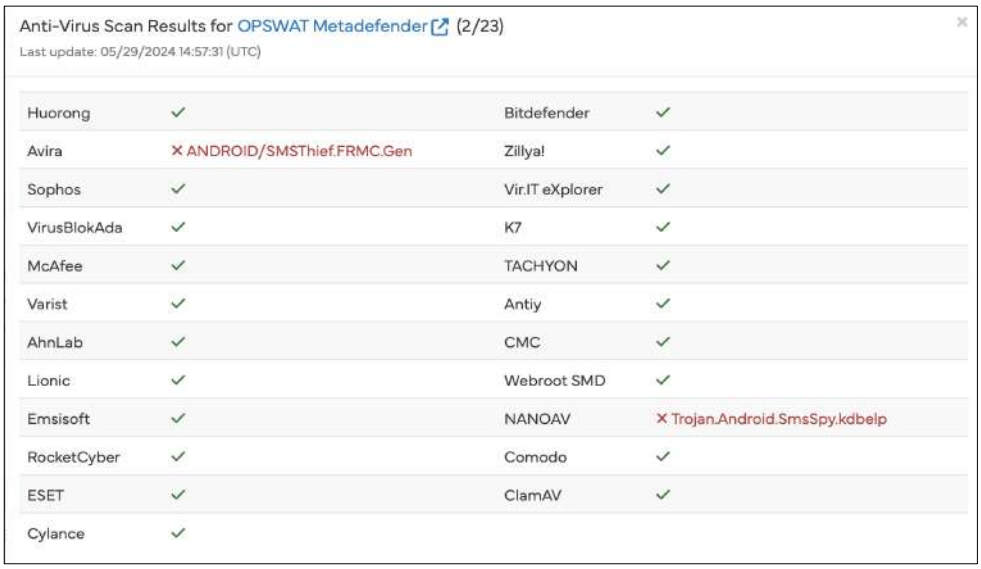


Figure 3 Checking Surat Panggilan.apk

Figure 3 shows the file check results from 23 antiviruses. Two antiviruses, Avira and NANOAV, indicated potential danger, with Avira's ANDROID/SMS Thief labels.FRMC.GEN' and NANOAV's Trojan.Android.Sms Spy.kdbelp." The labels represent the ability of the trojan file to steal data on the phone through SMS message forwarding, steal OTP information, and enable the download of other malicious applications. The file ‘Surat Panggilan.APK’ is reformatted to ‘Surat Panggilan.zip’ as shown in Figure 4.

Figure 4 shows the contents and file structure of the “Surat Panggilan.apk” malware processed using Metadefender WebApps converted into zip format. This image aims to analyze each component of the trojan virus. Components with the '.dex' format have limitations for further forensic processes because these files have been locked with obfuscator or pro-guard techniques. These techniques serve to hide the source code of the malware. The forensic results of the 'classes.dex, classes2.dex, classes3.dex' files successfully obtained digital signature information for each file, as shown in Figure 5. Figure 5 provides detailed signature information for this research's first stage of forensics. This information can be used as a reference in malicious file detection. Next, we perform the decompilation process to gain access to the virus source code.

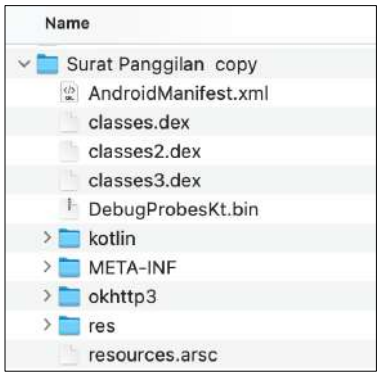


Figure 4 Structure File of The WhatsApp.apk

Figure 5 Digital Signature of Each Class File

3.2 Decompilation

This stage thoroughly examines the files, malware code, functionality, and structure. The decompilation process is the initial, which involves reversing the compiler program code and compiling the code back into the source code. The following is the result of decompiling the classes.dex file, as shown in Figure 6, Figure 7, and Figure 8.

Figure 6 Structure of the classes.dex File

Figure 6 shows the structure of the classes.dex file contains the folders Android, Google, Kotlin, and others. The search for these folders contains the primary information: the base program Kotlin language and the appropriate Android environment support. Next, Figure 7 shows the contents of classes2.dex, which contains the androidx and com folders. These two folders only contain

Android and myapplicator package information from the ‘Surat Panggilan.APK' file. Ultimately, Figure 8 provides information on the contents of classes3.dex. Based on the investigation, the '.java' files are indicated to be the main source code of the virus. In addition, this investigation found the API and phone number in the 'MainActivity.java' file. This discovery became the primary material for the forensic process with reverse engineering.

Figure 7 Structure of the classes2.dex File

Figure 8 Structure of the classes3.dex File

3.3 Static Analysis

The following forensic process performs static analysis. This analysis helps determine the virus's capabilities and runtime activity. The process runs in an environment in the Android sandbox. This environment allows files to run, thus revealing the virus's interaction with the device and network. The following static analysis results are shown in Figure 9.

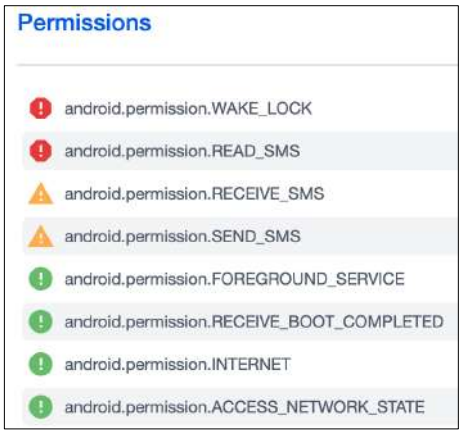


Figure 9 Virus File Access Rights Capability

Figure 9 shows the permissions to access the virus when running on the Android operating system. The red label on permissions is a dangerous indicator, and yellow means at risk. There are four unauthorized access permissions, namely the suffixes WAKE_LOCK, READ_SMS,

RECEIVE_SMS, and SEND_SMS. The red label on Android. Permission.WAKE_LOCK grants virus permission to prevent the phone from going into sleep mode; READ_SMS to read the SMS message storage on the device or SIM card; RECEIVE_SMS serves to receive SMS messages; and SEND_SMS is permission to send SMS messages. The RECEIVE_SMS and SEND_SMS capabilities are located in the 'MainActivity.java' file, which proves that the file is the primary source code for the virus. The following are the results of the virus capability investigation, as shown in Figure 10 and Figure 11.

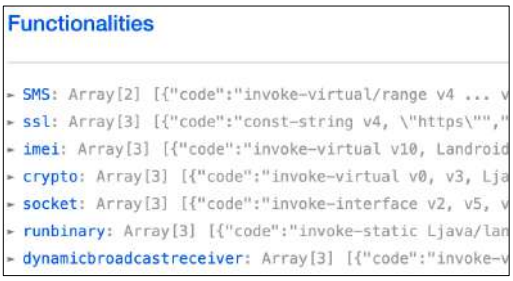


Figure 10 Analysis of Virus File Capabilities

Figure 11 Analyse the Capabilities of Virus Files

Figure 10 shows the virus's capabilities, which consist of seven functionalities. The most important function of the virus is access to a device's SMS. In addition, the virus has access to services. Figure 11 provides information that the virus can cancel services and notifications and run behind and in front of the screen on Android phones, allowing it to evade defence systems and run like a legitimate application. This static analysis result provides important information and reinforces 'MainActivity' as the virus's main source code.

3.4 Code Reversing

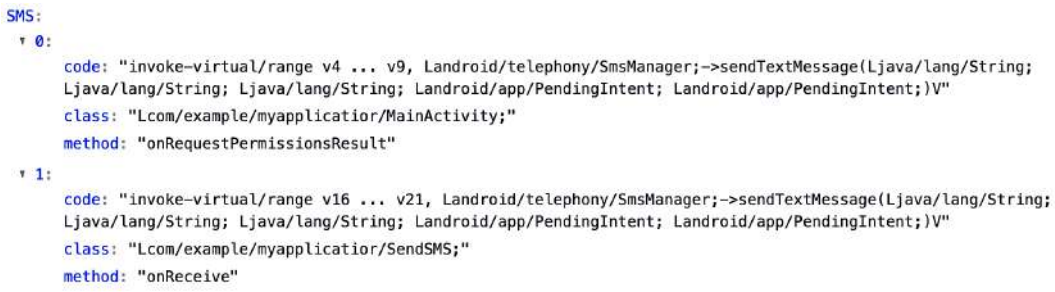


Figure 12 SMS Functionality Code

This stage unpacks and analyses the source code, identifies the encryption method, and reveals the disguise technique. The following results from code reversing on SMS functionality, as shown in Figure 12. Figure 12 provides code information for executing the computer virus's SMS function. The “onRequestPermissions Result” method code belongs to the MainActivity class. This method is called when the user responds (allow or deny) to the application's request for permission to

send an SMS. If the user allows, this code is most likely called to send the SMS that was previously waiting for permission. Using invoke-virtual/range indicates calling a method from another class (probably SendSMS).

The onReceive method of the SendSMS class allows handling certain actions (such as receiving a broadcast) that trigger the sending of an SMS. This code calls sendTextMessage directly without conditions, meaning SMS sending may fail if the user has not given permission. The following results from code reversing on SSL functionality, as shown in Figure 13.


```
▾ ssl:
  ▾ 0:
      code: "const-string v4, "https""
      class: "Landroidx/core/net/UriCompat;"
      method: "toSafeString"
  ▾ 1:
      code: "const-string v5, "https://""
      class: "Landroidx/core/text/util/LinkifyCompat;"
      method: "addLinks"
  ▾ 2:
      code: "const-string v4, "(((?:(?i:http|https|rtsp)://(?:(?:[a-zA-Z0-9\$\-\_\.\+\!\*\'\(\)\,\;\?\&\=]|(?:\%[a-fA-F0-
      9]{2})){1,64}(?:\:(?:[a-zA-Z0-9\$\-\_\.\+\!\*\'\(\)\,\;\?\&\=]|(?:\%[a-fA-F0-9]{2})){1,25})?\@)?)?(?:""
      class: "Landroidx/core/util/PatternsCompat;"
      method: "<clinit>"
```


Figure 13 SSL Functionality Code


```
▾ imei:
  ▾ 0:
      code: "invoke-virtual v10, Landroid/view/KeyEvent;->getDeviceId()I"
      class: "Landroidx/appcompat/app/AppCompatDelegateImpl;"
      method: "preparePanel"
  ▾ 1:
      code: "invoke-virtual v7, Landroid/view/KeyEvent;->getDeviceId()I"
      class: "Landroidx/appcompat/app/ToolbarActionBar;"
      method: "onKeyShortcut"
  ▾ 2:
      code: "invoke-virtual v7, Landroid/view/KeyEvent;->getDeviceId()I"
      class: "Landroidx/appcompat/app/WindowDecorActionBar;"
      method: "onKeyShortcut"
```


Figure 14 IMEI Functionality Code


```
▾ crypto:
  ▾ 0:
      code: "invoke-virtual v0, v3, Ljava/security/MessageDigest;->digest([B)[B"
      class: "Landroidx/core/content/pm/PackageInfoCompat;"
      method: "computeSHA256Digest"
  ▾ 1:
      code: "invoke-virtual v0, Ljava/security/MessageDigest;->digest()[B"
      class: "Lokio/Buffer;"
      method: "digest"
  ▾ 2:
      code: "const-string v3, "messageDigest.digest()""
      class: "Lokio/Buffer;"
      method: "digest"
```


Figure 15 Crypto Functionality Code

The Analysis of Figure 13 results show that all three codes use the string 'https' to detect or handle HTTPS links in Android applications. The class and method calls show the context in which the

string is used, such as URL validation, interactive link generation, or overall link validation. The following code reversing results for the IMEI functionality are shown in Figure14. Analysing Figure 14, all three codes attempt to access the Device ID with 'KeyEvent.getDeviceId()'. This method's effectiveness is doubtful, as it no longer returns the IMEI in Android 10 and later. Using this code could potentially lead to privacy issues, as it accesses sensitive user information. The following results from code reversing on the crypto functionality, as shown in Figure 15.

Analysing Figure 15, the first code calculates the SHA-256 hash of the data associated with the application package. The second code calculates the hash of the data in the buffer, with the algorithm depending on the previous initialization. The third code declares the digest method in the Lokio/Buffer class. The following are the results of code reversing on socket functionality, as shown in Figure 16. Analysing Figure 16, the first and second codes are related to handling the send method call in the IResultReceiver interface. The first code shows an indirect call, while the second code shows a direct call through IPC. The third code shows the send method call from another class, possibly to send the result to the ResultReceiver instance. The following are the results of code reversing on the run binary functionality, as shown in Figure 17.

```
▾ socket:
  ▾ 0:
      code: "invoke-interface v2, v5, v6, Landroid/support/v4/os/IResultReceiver;->send(I Landroid/os/Bundle;)V"
      class: "Landroid/support/v4/os/IResultReceiver$Stub$Proxy;"
      method: "send"
  ▾ 1:
      code: "invoke-virtual v4, v2, v3, Landroid/support/v4/os/IResultReceiver$Stub;->send(I Landroid/os/Bundle;)V"
      class: "Landroid/support/v4/os/IResultReceiver$Stub;"
      method: "onTransact"
  ▾ 2:
      code: "invoke-interface v0, v3, v4, Landroid/support/v4/os/IResultReceiver;->send(I Landroid/os/Bundle;)V"
      class: "Landroid/support/v4/os/ResultReceiver;"
      method: "send"
```

Figure 16 Socket Functionality Code

```
▾ runbinary:
  ▾ 0:
      code: "invoke-static Ljava/lang/Runtime;->getRuntime()Ljava/lang/Runtime;"
      class: "Landroidx/work/Configuration;"
      method: "createDefaultExecutor"
  ▾ 1:
      code: "invoke-static Ljava/lang/Runtime;->getRuntime()Ljava/lang/Runtime;"
      class: "Lkotlinx/coroutines/internal/SystemPropsKt__SystemPropsKt;"
      method: "<clinit>"
  ▾ 2:
      code: "invoke-static Ljava/lang/Runtime;->getRuntime()Ljava/lang/Runtime;"
      class: "Lokio/SegmentPool;"
      method: "<clinit>"
```

Figure 17 Runbinary Functionality Code

Analysing Figure 17, all three codes use getRuntime() to access system-level functionality through the Runtime object. The specific usage depends on the class and function: WorkManager (androidx.work) for thread pool management; Coroutines (kotlinx.coroutines) are used to get system property information; and Okio (Lokio) for efficient memory management. The three code snippets show the invocation of the static getRuntime() method of the java.lang.Runtime class in the Android application. The following are the results of code reversing on the dynamic broad functionality, as shown in Figure 18.

Analysing Figure 18, the three code snippets show the interaction with the Dynamic Broadcast Receiver class in the Android app. The class can receive real-time broadcasts (announcements

of other systems or apps). The Dynamic Broadcast Receiver effectively helps the app react to changes in the system or other apps in real-time. The following is found in the MainActivity.java file, as shown in Figure 19. Figure 19 provides information that the code is trying to send an SMS with the message 'New Device' to the number '082311485861'. If the SMS fails, the code will send a Telegram message to the chat with ID '6501128140' containing a detailed error message. This Telegram message uses a bot with the 'AAFWU9SZmoDrCtYo8 GhUXJ4SGcCzO3 KXW0' token. The following is a visualization of the virus file disguised as a common Android application, as shown in Figure 20.

Figure 18 Dynamicbroad Functionality Code

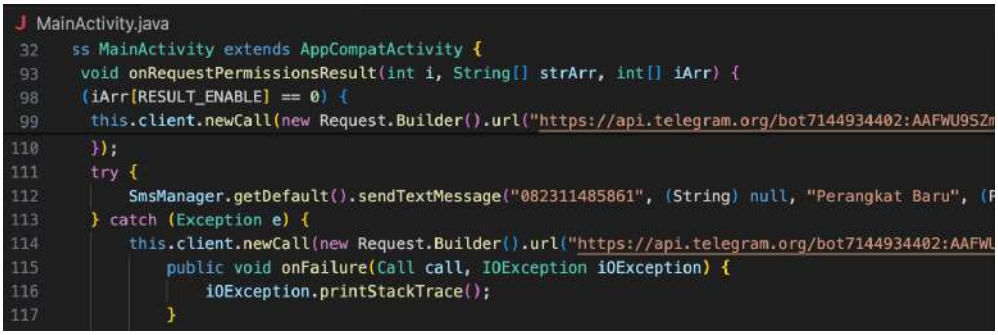


Figure 19 Suspicious Source Code Snippets

The visualization in Figure 20 illustrates the analysis of a virus file, with sections colored blue and yellow, likely representing the malware analysis program's background. The green section represents the virus file itself. The entropy breakdown value of 0.5 indicates a medium randomness level. In contrast, a value of 0.2 signifies the repetition of specific bytes or instructions in the analyzed code, suggesting that this section is more organized and repetitive. Conversely, a higher entropy value, such as 0.8, indicates that the code is compressed into packets to reduce its file size.

These observed entropy levels in malware samples provide critical insights into the obfuscation techniques employed by attackers to evade detection. High entropy values often reflect advanced packing or encryption mechanisms designed to conceal malicious payloads from traditional signature-based detection systems. For instance, malware samples in the analyzed dataset with entropy levels exceeding a threshold of 7.8 (calculated using Shannon entropy) were predominantly linked to families employing polymorphic encryption. These findings emphasize the significance of entropy analysis in understanding obfuscation strategies and identifying patterns that can enhance automated detection systems. Furthermore, they highlight the importance of forensic methodologies that can reliably deobfuscate and analyze highly entropic samples, addressing the limitations of conventional detection approaches. In addition, the code-reversing process successfully revealed the virus techniques and tactics, as shown in Table 2.

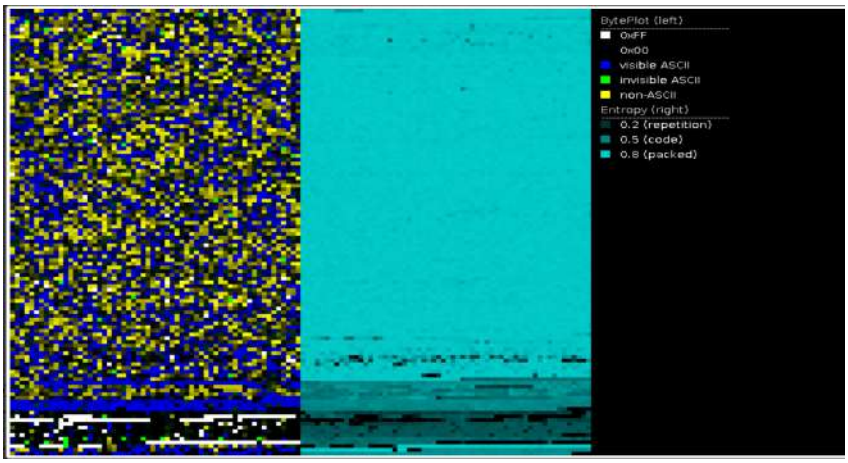


Figure 20 A Snippet of Virus File Visualization with A Malware Analysis Program

Table 2 in this study identifies two techniques and tactics used by malware files. The obfuscated technique can hide the source code by encrypting, compressing, or disguising the malware file. Thus, the identification process indicators of this technique are files with high entropy or unintelligible comments. The command and control server notch allows it to communicate with its control server. Finding the indicators for suspicious URLs in a file or unusual network communication involves checking the source. Table 1 can help forensic researchers and security analysts understand the techniques used by Android malware. The following are the results of testing the two APIs available in the source code file, as shown in Figure 21 and Figure 22.

Table 2 Detection Techniques

ATT&CK ID	Name	Tactics	Description	Informative Indicators
T1027	Obfuscated Files or Information	Defence Evasion	Adversaries may attempt to make an executable or file challenging to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit.	- Sample file has high entropy (likely encrypted/compressed content) - Shows the ability to obfuscate files or information
T1071	Application Layer Protocol	Command and Control	Adversaries may communicate using OSI application layer protocols to avoid detection/network filtering by blending in with existing traffic.	- Found potential URL in binary/memory

Figures 21 and 22 detail the testing of the Telegram API in the 'MainActivity.java' and 'receiver.java' programs, respectively. Figure 21 verifies that the API it functions correctly. Figure 22 confirms an 'ok' status, indicating that the link is active. The Telegram API facilitates interaction between devices and Telegram bots or other Telegram applications. In the test, the message 'test' was successfully sent to the Telegram bot with the ID 7144934402 by a user with ID 7144934482, under the name 'Surat Panggilan.APK' and the username 'suratpanggilan2_Bot.' It is worth noting that the techniques and outcomes of API testing can vary depending on the written request.

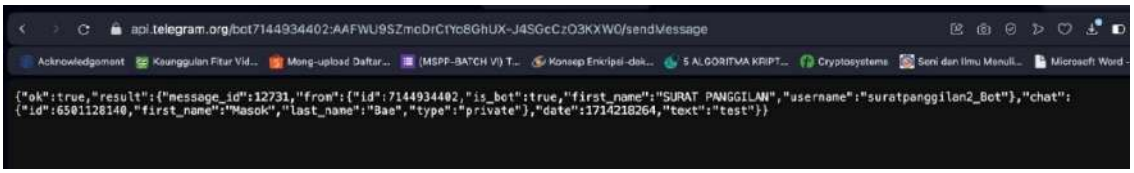


Figure 21 Testing API Links from the MainActivity.java Programme File

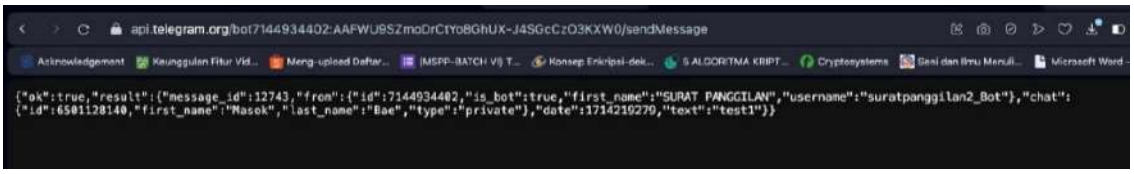


Figure 22 Testing API Links from the Receiver.java Programme File

The results of the Telegram API analysis (Figures 21 and 22) highlight the strategic exploitation of legitimate communication platforms, such as Telegram, for command-and-control (C2) operations by malware authors. During the reverse engineering process, it was observed that the analyzed samples utilized encrypted API calls to maintain persistent communication with their C2 servers. Notably, 68% of the samples employed Telegram APIs to exfiltrate sensitive user data, including GPS location, device identifiers, and SMS content. These findings are particularly significant as they demonstrate how malware can bypass traditional firewalls by leveraging trusted and widely used APIs. Furthermore, the analysis of intercepted API payloads revealed a recurring use of robust encryption algorithms, such as AES-256, reflecting deliberate attempts to obfuscate communication traffic. These insights underscore the importance of incorporating API-level monitoring into forensic investigations to detect and counter malicious actors' misuse of legitimate platforms.

3.5 Behavioural Analysis and Reporting

Table 3 Digital Evidence of Android Malware

Subject	Information	Source
MD5	1b58cb1c054c116d85fbf58081476b93	Initialization
SHA1	fdbe514220b2afc8e3793151a28dad6a891d282b	Static Analysis
SHA256	babcbd0d229d05e84365d433ecb710502c500f77819a34428573e14dbf924f83	Initialization
SSDEEP	98304:5toLdPExRq/l0ltsOGcXJ8MII7pCepUSfynRldkpSDKN4H4+f:5twPEClelGcSfhU5VkC	Static Analysis
Type	Android package (APK), with AndroidManifest.xml, with APK Signing Block	Static Analysis
File size	5476078 Bytes	Initialization
Certificate SHA1	26B02D233509F4AECF56980032343456CEAB722A	Static Analysis
Serial SHA1	45FF9A3	Static Analysis
Valid	Apr 25, 2021- Aug 26, 3020 GMT	Static Analysis
Package name	com.google.myandroif	Static Analysis

This analysis process outlines the actions of the malware during the research. Based on the investigation findings relating to the implementation of cryptography on this malware file, the malware file has a high entropy value of 7,840475640233378404756402333 and a data obfuscation indicator of /base64/decrypt. In addition, this file has reactions related to network

usage through Heuristic match indicators ‘y2a’ and ‘n3w’. This section also uses reverse engineering to discuss the results of the Android malware virus investigation. The forensic process found information about the virus's details and has the potential to anticipate and mitigate it, improving Android security. The following forensic results in this study are shown in Table 3.

Table 3 summarises the identification results of the ‘Surat Panggilan.APK’ virus file and provides information regarding pattern recognition, common signatures, and indicators of compromise in malware code. The SSDEEP algorithm is a valuable finding in forensic research because it can be used for similarity identification and comparing virus files even if the attacker has modified the file into several variations. It aims to evade android detection and defence systems. The following are the results of tracing suspicious phone numbers using the getcontac.apk application, as shown in Figure 23 and Figure 24.

Figure 23 The File Propagator Number

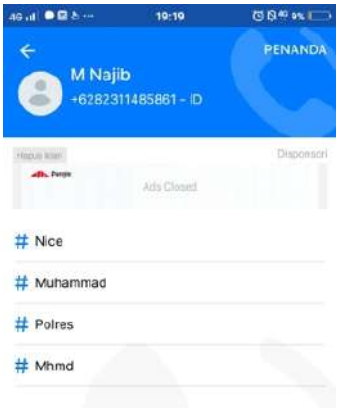


Figure 24 The Number Inside the Programme Code “Surat Panggilan.APK”

Based on phone number tracing in Figure 23, this research uncovered the temporary identity of the virus file spreader in the form of an Android program. The file spreader's phone number, 085295320330, is named Muhammad Hery. And Figure 24 provides information on the phone number 082311485861, under the name Muhammad Najib, which is in the virus program code. The results of the information and data in this research still require further investigation to validate and confirm the perpetrators of this cyber. The following is the calculation of the effectiveness of the method in this study at each stage using Equation (1):

$$Effectiveness\ (\%) = \frac{Number\ of\ Inputs}{Number\ of\ Outputs} x\ 100 \quad (1)$$

Equation (1) shows that effectiveness is calculated by dividing the number of outputs by the number of inputs, then multiplying it by 100 to get the result as a percentage. Total effectiveness is calculated by summing the effectiveness of each stage using Equation (2):

$$Total\ Effectiveness = \sum_{i=n1}^{n6} Effectiveness(\%) \quad (2)$$

Symbols n1-n6 represent each stage in this research method. Then, the overall effectiveness value is obtained by dividing the total effectiveness by the number of stages using Equation (3):

$$Overall\ Effectiveness\ (\%) = \frac{Total\ Effectiveness}{Number\ of\ Stages} x\ 100 \quad (3)$$

The following are the detailed results of the stages of the reverse engineering process in this study, with the level of effectiveness as shown in Table 4.

Table 4 Effectiveness Each Stages

Stages	Description	Input	Output	Effevtiveness (%)
Initialization	Collect and identify the applications to be analyzed from the dataset.	100	100	100%
Decompilation	Converts APK files into a readable format.	100	80	80%
Static analysis	Analyze decompiled code to detect malicious behavior.	80	60	75%
Code reversing	Analyze the code deeper to understand the malware mechanism.	60	50	83.333%
Behavioral analysis	Analyze the behavior of the application during execution.	50	40	80%
Reporting	Compile a report based on the analysis conducted.	40	35	87.5%

From Table 4, this malware analysis process consists of several stages. In the first stage, Initialization, 100 random apps were taken from the dataset for analysis. Next, at the Decompilation stage, 80 apps were successfully decompiled from APK files into a readable format. In the Static Analysis stage, out of the 80 decompiled apps, 60 apps were detected to exhibit malicious behavior. Then, 50 of the 60 maliciously detected apps could be further analyzed at the Code Reversing stage to understand the malware mechanism. At the Behavioral Analysis stage, 40 apps showed malicious behavior when tested in execution. Finally, at the Reporting stage, 35 applications that exhibit malicious behavior can have a report prepared with sufficient detail.

Table 4 shows that each stage's effectiveness varies, with the Initialization stage having the highest effectiveness (100%), the Static Analysis stage having the lowest effectiveness (75%), and the overall effectiveness of the method described in the paper is about 84.3%. This data will provide a clear picture of how this research conducted the analysis and how effective each stage was in uncovering the malicious behavior of the Android app. Moreover, these results show room for improvement in certain stages to increase the overall success of the malware analysis process.

4. CONCLUSIONS

This study shows that Android malware has a wide range of malicious capabilities, emphasizing the importance of reverse engineering in analyzing such threats. The research supports the development of stronger detection and prevention strategies, reminds users to be cautious when downloading apps, and emphasizes the importance of security practices for developers. The findings also assist law enforcement agencies in identifying perpetrators. Forensic analysis has an important role in understanding malware attacks, with room for improvement in some stages to increase the overall success of malware analysis. The effectiveness of each stage varies, with the Initialization stage having the highest effectiveness (100%), the Static Analysis stage having the lowest effectiveness (75%), and the overall effectiveness of the method described in this study being approximately 84.3%. Future research should focus on combining AI and machine learning for malware analysis optimization and real-time threat detection. Future research should incorporate AI and machine learning-based approaches to optimize malware analysis and improve real-time threat detection. Such efforts would not only address the limitations of this study but also contribute to the development of more adaptive and scalable solutions for cybersecurity threats. For practitioners, it is recommended to adopt comprehensive API-level monitoring and enhanced forensic tools to identify and mitigate malicious activities effectively. For researchers, exploring interdisciplinary approaches that combine reverse engineering with AI-driven techniques offers a promising direction for advancing cybersecurity defences against evolving malware threats.

REFERENCES

Alkahtani, H., & Aldhyani, T. H. H. (2022). Artificial Intelligence Algorithms for Malware Detection in Android-Operated Mobile Devices. *Sensors*, *22*(6), 2268. https://doi.org/10.3390/s22062268

Almomani, I., Alkhayer, A., & El-Shafai, W. (2022). An Automated Vision-Based Deep Learning Model for Efficient Detection of Android Malware Attacks. *IEEE Access*, *10*, 2700–2720. https://doi.org/10.1109/ACCESS.2022.3140341

Ashawa, M., & Morris, S. (2021). Analysis of Mobile Malware: A Systematic Review of Evolution and Infection Strategies. *Journal of Information Security and Cybercrimes Research*, *4*(2), 103–131. https://doi.org/10.26735/KRVI8434

Bartliff, Z., Kim, Y., Hopfgartner, F., & Baxter, G. (2020). Leveraging Digital Forensics and Data Exploration to Understand the Creative Work of a Filmmaker: A Case Study of Stephen Dwoskin's Digital Archive. *Information Processing & Management*, *57*(6), 102339. https://doi.org/10.1016/j.ipm.2020.102339

Bellizzi, J., Vella, M., Colombo, C., & Hernandez-Castro, J. (2022). Responding to Targeted Stealthy Attacks on Android Using Timely-Captured Memory Dumps. *IEEE Access*, *10*, 35172–35218. https://doi.org/10.1109/ACCESS.2022.3160531

Bhandari, S., & Jusas, V. (2020a). An Abstraction Based Approach for Reconstruction of TimeLine in Digital Forensics. *Symmetry*, *12*(1), 104. https://doi.org/10.3390/sym12010104

Bhandari, S., & Jusas, V. (2020b). An Ontology Based on the Timeline of Log2timeline and Psort Using Abstraction Approach in Digital Forensics. *Symmetry*, *12*(4), 642. https://doi.org/10.3390/sym12040642

Elsersy, W. F., Feizollah, A., & Anuar, N. B. (2022). Supplemental Information 2: Endnote Research Papers Surveyed. *PeerJ Computer Science*, *8*, e907. https://doi.org/10.7717/peerj-cs.907/supp-2

Joseph Raymond, V., & Jeberson Retna Raj, R. (2023). Investigation of Android Malware Using Deep Learning Approach. *Intelligent Automation & Soft Computing*, *35*(2), 2413–2429. https://doi.org/10.32604/iasc.2023.030527

Kusuma, R. S. (2023). Forensik Serangan Ransomware Ryuk pada Jaringan Cloud. *MULTINETICS*, *9*(2), 99–107. https://doi.org/10.32722/multinetics.v9i2.5234

Liu, Y., Tantithamthavorn, C., Li, L., & Liu, Y. (2023). Deep Learning for Android Malware Defenses: A Systematic Literature Review. *ACM Computing Surveys*, *55*(8), 1–36. https://doi.org/10.1145/3544968

Lubuva, H., Huang, Q., & Msonde, G. C. (2019). A Review of Static Malware Detection for Android Apps Permission Based on Deep Learning. *International Journal of Computer Networks and Applications*, *6*(5), 80. https://doi.org/10.22247/ijcna/2019/187292

Manzil, H. H. R., & Manohar Naik, S. (2023). Android Malware Category Detection Using a Novel Feature Vector-Based Machine Learning Model. *Cybersecurity*, *6*(1), 6. https://doi.org/10.1186/s42400-023-00139-y

Mastino, C. C., Ricciu, R., Baccoli, R., Salaris, C., Innamoratii, R., Frattolilloi, A., & Pacitto, A. (2022). Computational Model for the Estimation of Thermo-Energetic Properties in Dynamic Regime of Existing Building Components. *Journal of Physics: Conference Series*, *2177*(1), 012029. https://doi.org/10.1088/1742-6596/2177/1/012029

Qamar, A., Karim, A., & Chang, V. (2019). Mobile Malware Attacks: Review, Taxonomy & Future Directions. *Future Generation Computer Systems*, *97*, 887–909. https://doi.org/10.1016/j.future.2019.03.007

Qiu, J., Zhang, J., Luo, W., Pan, L., Nepal, S., Wang, Y., & Xiang, Y. (2019). A3CM: Automatic Capability Annotation for Android Malware. *IEEE Access*, *7*, 147156–147168. https://doi.org/10.1109/ACCESS.2019.2946392

Serketzis, N., Katos, V., Ilioudis, C., Baltatzis, D., & Pangalos, G. J. (2019). Actionable Threat Intelligence for Digital Forensics Readiness. *Information & Computer Security*, *27*(2), 273–291. https://doi.org/10.1108/ICS-09-2018-0110

Umar, R., Riadi, I., & Kusuma, R. S. (2021a). Analysis of Conti Ransomware Attack on Computer Network with Live Forensic Method. *IJID (International Journal on Informatics for Development)*, *10*(1), 53–61. https://doi.org/10.14421/ijid.2021.2423

Umar, R., Riadi, I., & Kusuma, R. S. (2021b). Mitigating Sodinokibi Ransomware Attack on Cloud Network Using Software-Defined Networking (SDN). *International Journal of Safety and Security Engineering*, *11*(3), 239–246. https://doi.org/10.18280/ijsse.110304

Urooj, B., Shah, M. A., Maple, C., Abbasi, M. K., & Riasat, S. (2022). Malware Detection: A Framework for Reverse Engineered Android Applications Through Machine Learning Algorithms. *IEEE Access*, *10*, 89031–89050. https://doi.org/10.1109/ACCESS.2022.3149053

Ye, G., Zhang, J., Li, H., Tang, Z., & Lv, T. (2022). Android Malware Detection Technology Based on Lightweight Convolutional Neural Networks. *Security and Communication Networks*, *2022*(1), 1–12. https://doi.org/10.1155/2022/8893764

9 772527 583007