Journal of Applied Informatics Research Vol 01 No 01 (2025) pp. 30~44

Journal of Applied Informatics Research
Fakultas Vokasi, Kampus Unesa 1, Ketintang, Surabaya, Indonesia
Website: https://journal.unesa.ac.id/index.php/jair/ | E-mail: jair@unesa.ac.id

Enhancing Clickbait Headline Identification Performance Without Preprocessing Through Feature Reduction and Sentiment Analysis

Anisa Nur Azizah[1], Misbachul Falach Asy'ari[2], Moch Deny Pratama[3], Dimas Novian Aditia Syahputra[4], M Adamu Islam Mashuri[5], Binti Kholifah[6], Rifqi Abdillah[7], Adinda Putri Pratiwi[8], Dina Zatusiva Haq[9]

[1]Department of Informatics Engineering, Universitas Wijaya Putra, Indonesia
[2,8]Department of Informatics Engineering, Institut Teknologi Sepuluh Nopember, Indonesia
[3,4,5,6]Department of Informatics Management, Universitas Negeri Surabaya, Indonesia
[7]Department of Informatics Engineering, Universitas Negeri Surabaya, Indonesia
[9]Department of Informatics, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

[1]anisanurazizah@uwp.ac.id, [2]misbachulfalach@gmail.com, [3]mochpratama@unesa.ac.id, [4]dimassyahputra@unesa.ac.id, [5]mmashuri@unesa.ac.id, [6]bintikholifah@unesa.ac.id, [7]rifqiabdillah@unesa.ac.id, [8]adindaputripratiwi97@gmail.com, [9]dinaza.if@upnjatim.ac.id

ARTICLE INFORMATION

Article history:

Received July 28, 2025
Revised July 29, 2025
Accepted July 30, 2025

Keywords:

Clickbait Detection;
Feature Reduction;
Sentiment Analysis;
Embedding Technique;
Machine Learning.

ABSTRACT

This study addresses the challenge of identifying clickbait headlines without relying on conventional text preprocessing, which can be resource-intensive and may degrade contextual integrity. To enhance detection performance, we examine three feature extraction methods: TF-IDF, Word2Vec, and Headline2Vec an embedding technique designed for short texts like headlines. These features are optimized using feature selection algorithms, including Pearson Correlation Coefficient (PCC), Neighborhood Component Analysis (NCA), and Relief, to reduce dimensionality and enhance relevant signal retention. Sentiment polarity is also integrated as a complementary feature. A comparative evaluation is conducted using several machine learning classifiers, namely Support Vector Classifier (SVC), Random Forest, LightGBM, and XGBoost, across all combinations of feature extraction and selection methods. Results show that the optimal configuration Headline2Vec with Relief and SVC achieves the highest accuracy at 94.40%, outperforming other approaches. This demonstrates the effectiveness of combining semantic vectorization and feature selection for clickbait detection in the absence of traditional preprocessing. The findings support the development of streamlined and scalable classification models capable of maintaining high accuracy while reducing preprocessing overhead, making the proposed method particularly suitable for real-time and large-scale content moderation and news verification systems.

1. INTRODUCTION

The rapid growth of internet access and social media usage has drastically changed the patterns of information dissemination and consumption worldwide, including in Indonesia. According to a report by the Indonesian Ministry of Communication and Informatics (Kominfo), internet users in the country increased by approximately 11% in one year, from 175.4 million to 202.6 million users [1]. This growth has significantly impacted the media landscape, accelerating the shift from traditional print journalism to digital news platforms and online web portals.

Digital media offers several substantial advantages, such as wider accessibility, instant updates, lower publication costs, and compatibility with mobile devices. As a result, users can now consume news anytime and anywhere, increasing competition among news portals for readers' attention. To attract user engagement and increase click-through rates (CTR), many online news platforms have adopted clickbait strategies, particularly in the construction of news headlines [2]. Clickbait refers to the deliberate use of sensational, ambiguous, or misleading headlines designed to arouse curiosity while concealing key facts within the article's content [3]. These headlines often use hyperbole, emotionally charged language, or incomplete information to encourage users to click. While clickbait can increase visibility and advertising revenue, it also raises concerns about credibility, user trust, and the integrity of journalistic practices. The proliferation of such content has led to public dissatisfaction, as users increasingly feel deceived or misinformed. This duality highlights the need for automated systems capable of detecting and classifying clickbait content accurately and in real-time.

Previous research has explored various methods for addressing clickbait detection using machine learning and deep learning approaches. Siregar et al. [2] implemented a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) units to detect clickbait in Indonesian news headlines, achieving an accuracy of 83%. William et al. [4] analyzed the Click-ID dataset, consisting of 15,000 headlines from 12 major Indonesian news portals, and applied Bi-LSTM with data preprocessing, resulting in an average accuracy of 77.8%. Meanwhile, Kaothanthong et al. [5] used a large-scale Thai news headline dataset and applied feature extraction using TF-IDF and Headline2Vec, followed by classification using Naive Bayes, SVM, and MLP. The best result 93.89% accuracy was obtained using Headline2Vec with MLP. These studies emphasize the crucial role of feature representation and preprocessing in improving classification performance. However, a persistent challenge in text-based classification is the high feature dimensionality, which arises from the large vocabulary and sparse nature of text data [6]. High-dimensional data can increase model complexity, reduce generalization, and prolong training time. To address these issues, feature selection and dimensionality reduction techniques are used to remove irrelevant or redundant features while retaining informative patterns. Several established algorithms, such as Relief [8], Pearson Correlation Coefficient (PCC) [9], and Neighborhood Component Analysis (NCA) [10], have proven effective in improving model accuracy and efficiency.

In response to these challenges, this study proposes a comprehensive approach for identifying clickbait headlines without relying on conventional text preprocessing techniques. We evaluate the performance of three feature extraction methods: TF-IDF, Word2Vec, and Headline2Vec, combined with three feature reduction algorithms: PCC, NCA, and Relief. Furthermore, sentiment polarity is included as an additional feature, with the hypothesis that emotional tone contributes to clickbait propensity. Several machine learning classifiers, including Support Vector Classifier (SVC), Random Forest, LightGBM, and XGBoost, are used to evaluate the best-performing combination. By forgoing traditional preprocessing steps, this study aims to demonstrate that semantic vectorization and precise feature reduction can effectively detect clickbait, while simultaneously improving the system's computational efficiency and scalability. These findings are expected to contribute to the development of real-time clickbait detection systems for practical use in content moderation, digital journalism, and misinformation prevention frameworks.

2. RESEARCH METHODS

2.1. Pre-Processing

Pre-processing is a process that aims to make data more structured and has the same form so the model can analyze. Pre-processing methods that we used in this research are lemmatization, lowercase, stopwords, and removing punctuation and number.

2.1.1. Lemmatize

Normalization using vocabulary and word morphology analysis, lematize aims to remove inflectional endings and return to the basic form of the word.

Table 1. Lemmatization of Word and Phrase Variants into "Fall down"

Original Word	Lemmatized Word
Fall down	Fall down
fall	Fall down
falling	Fall down
dropping	Fall down
fall	Fall down

Original Word	Lemmatized Word
fall	Fall down
fall	Fall down
Drop	Fall down

2.1.2. Lowercase

Lowercase is a method that makes all letters lowercase in order to get consistent output results.

2.1.3. Stopword Removal

Stopword is to eliminate common words that often appear and are considered to have no meaning. Stopword examples are “yang”, “at”, “to”, “from”, and etc.

2.1.4. Removing Punctuation and Number

Removes punctuation and numbers in a sentence to make data processing easier.

2.2. Feature Extraction

2.2.1. Word2Vec

Word2Vec is a text processing using 2 hidden neural layers to factor words. The input is a text corpus, and the output is a collection of vectors, the feature vectors represent the corpus data words. Word2vec has 2 model architectures for generating word representations, continuous bag-of-word (CBOW) and continuous skip gram. The difference between these two architectures is that CBOW predicts the model based on the context of the word, the order in the word cannot affect the predictor, while the continuous skip gram predicts based on the word between the previous and following words.

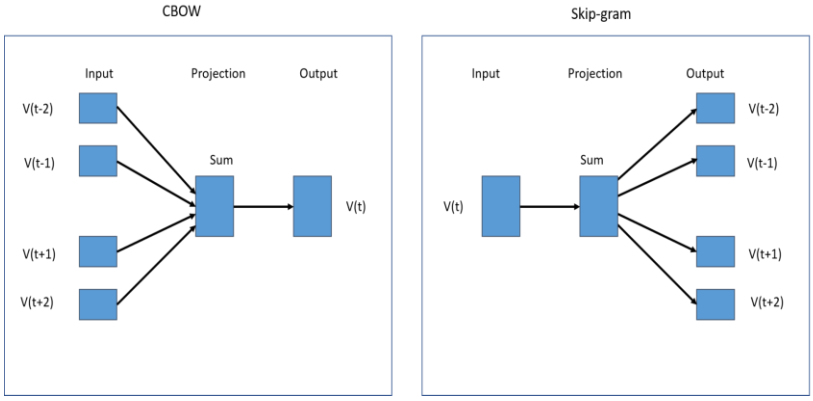

Figure 1. CBOW and Skip-gram Architectures in Word2Vec

2.2.2. TF-IDF

The TF-IDF algorithm is to calculate the weight of each word that appears most often in a text. The weights will be mapped into the vector n. TF weighting is done by calculating t (term) on d (document) (1) The weighting calculates the logarithm of the ratio of the number of documents in the corpus to the number of documents that have t(term) (2). The TF-ID value is obtained by multiplying the two.

The TF-IDF algorithm calculates the weight of each word that appears most often in a text. The weights are mapped into a vector. TF weighting is calculated by:

$$TF - IDF(t,d) = TF(t,d) \times log\left(\frac{N}{DF(t)}\right) \quad (1)$$

where TF(t,d) is the frequency of term ttt in document ddd, DF(t) is the number of documents containing term ttt, and NNN is the total number of documents in the corpus.

2.2.3. Headline2Vec

Headline2Vec is a method derived from the convolutional neural network (CNN) algorithm architecture. [paper thailand] The architecture of this method consists of two main parts, the first is feature extraction and hypertunning is used using CNN.

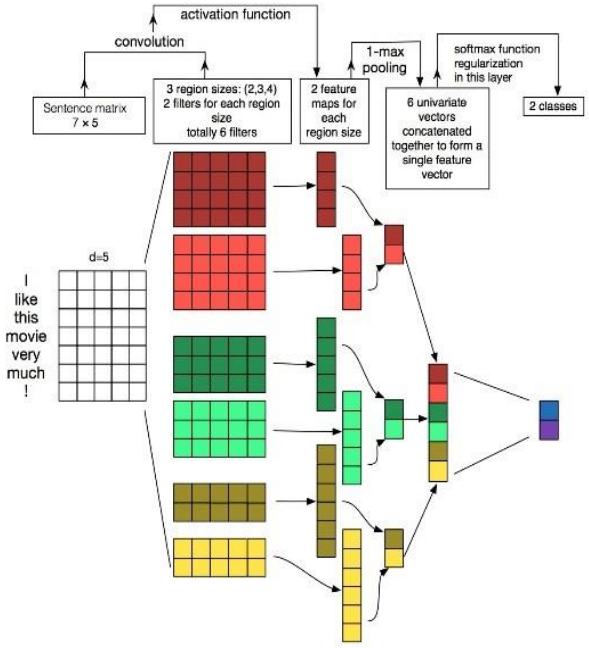

Figure 2. Architecture of the Convolutional Neural Network (CNN) used in Headline2Vec for feature extraction from text sequences.

2.3. Feature Reduction

2.3.1. Pearson Correlation Coefficient

Pearson Correlation Coefficient (PCC) is a correlation used to measure the strength and direction of the linear relationship of two variables. The independent variable and the dependent variable are interval scale. Correlation is based on a scale of 1 to -1, if it is close to a value of 1, the higher the positive correlation and will move simultaneously, if it is close to the value of -1, the value will be negative and the motion will be in the opposite direction, if the value is 0 then there is no correlation between the 2 variables.

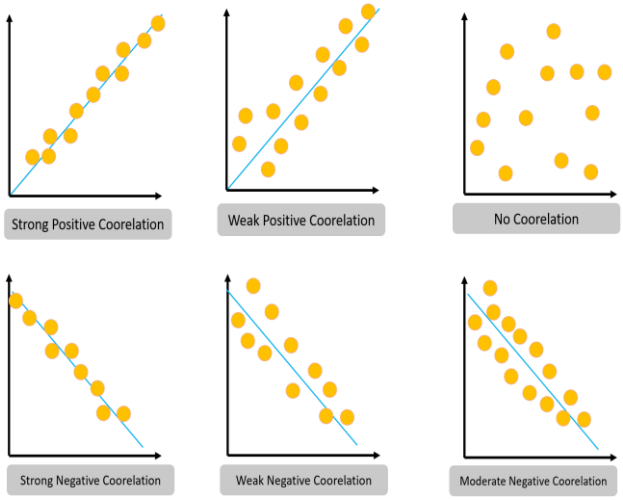

Figure 3. Visualization of different correlation patterns ranging from strong positive to strong negative correlation using Pearson Correlation Coefficient (PCC).

2.3.2. Neighborhood Component Analysis

Neighborhood Component Analysis (NCA) is a feature reduction method that uses the Mahalanobis distance metric found in the Supervised K-nearest neighbor (KNN) method to minimize leave-one-out (LOO) misclassification in training data. [1] The distance of the mahalanobis from the observation variables xi and xj is defined as follows:

$$D_M(x_i, x_j) = \sqrt{(x_i - x_j)^T W (x_i - x_j)} \quad (2)$$

Where w1 is the first feature weight. The above comparison will be used to obtain the highest LOO accuracy from the existing dataset.

NCA is different from Principal Component Analysis (PCA) which will eliminate some information because the raw data will be remapped into a simpler form or Sequential Feature Selection (SFS) which in some cases cannot remove unnecessary features after adding other features. NCA is not subject to data conditions and also will not change the existing data structure so that the information contained therein will remain intact. [2]

2.3.3. Relief

The relief algorithm is an algorithm inspired by instance-based learning. [3] This algorithm uses a proxy statistical variable for each feature to estimate the weight of each feature. This estimate is based on two things, the quality and relevance of each feature to be selected. This algorithm will assess each feature and assign a weight value to each feature. The weights to be given are in the range -1 to 1, where the value -1 is the worst variable representation and 1 is the best variable representation.

2.4. Classification Model

2.4.1. Random Forest

Random forest is a collection of decision trees that are used to classify data into a class. Random forests are supervised, so the more training data used, the more accurate the results will be. The decision tree contains three components, namely, decision nodes, leaf nodes, and root nodes. The decision tree algorithm will divide the dataset into branches, which will be separated into other branches. Branch splitting will continue until a leaf node is formed, and the leaf node can no longer be separated. Determination of the results of the classification is taken based on the voting results of the formed tree.

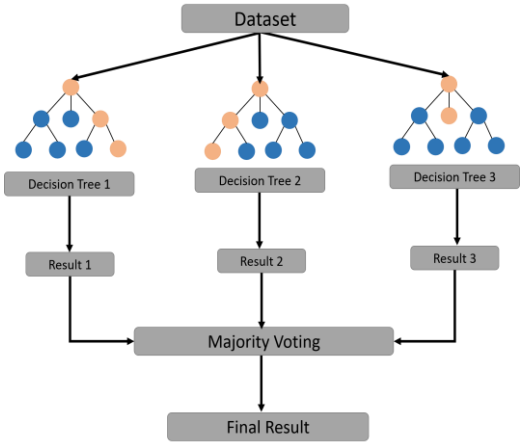

Figure 4. Decision Tree Structure

2.4.2. LightGBM

LightGBM is based on a decision tree algorithm. LightGBM uses a Gradient-based One-Side Sampling (GOSS) technique to filter the sample data to find the separator value, while the XGBoost algorithm uses a pre-filter and histogram-based process to find the best separator value.

LightGBM splits trees based on leaf nodes in contrast to other algorithms that grow trees. The LightGBM algorithm will select the leaf with the largest loss based on growth. Because the leaf nodes are fixed, it will produce less loss than other algorithms.

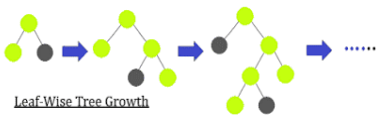

Figure 5. Illustration of leaf-wise tree growth in LightGBM

2.4.3. XGBoost

XGboost or eXtreme gradient Boost is an algorithm based on a decision tree which was developed from Gradient Boost with several additional processes. The process of trimming or proportional shrinking of leaf nodes is used to increase the generalizability of the model. the newton boosting process is a process to provide a direct route so that it does not require gradient descent. The parameter randomization process aims to reduce the correlation between trees so as to increase the strength of the ensemble algorithm.

Figure 6. Illustration of level-wise tree growth in XGBoost

2.4.4. SVC

Support Vector Classification is a classification method that can overcome several classification and regression problems with linear and non-linear with parameters in the form of support vectors that can be used to process data. Support Vector is designed for the problem of two object classes that have opposite properties.[6]

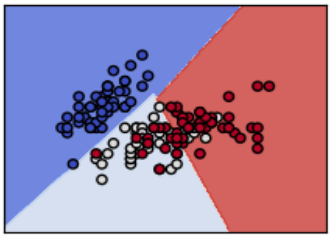

Figure 7. Support Vector Classification (SVC) Boundary

2.4.5. CNN

Convolutional Neural Network (CNN) is a type of neural network which is one of the Deep Learning algorithms for processing data. The application of convolutional neural networks is applied by using multi-layer perceptron (MLP) to classify images [7]. The use of CNN can be implemented for various processing and classifying data, both images and one-dimensional text [8].

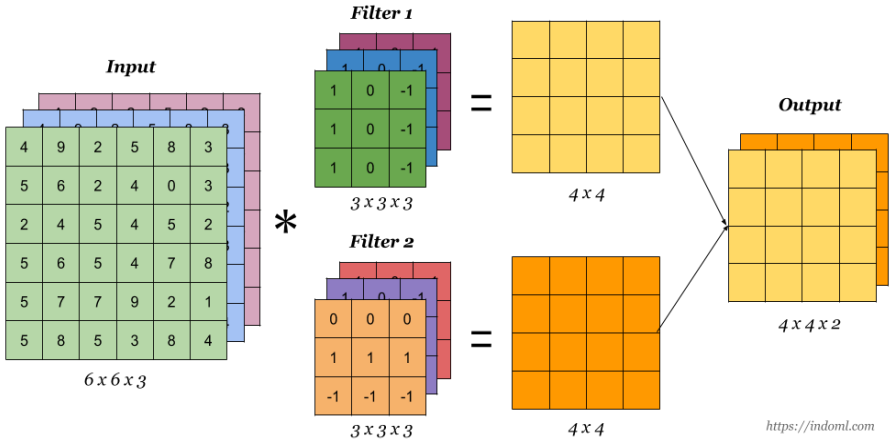

Figure 8. Convolution Operation in CNN

2.4.6. Bert

Bidirectional Encoder Representations from Transformers or BERT for short is a trained language representation model developed by researchers at Google AI Language in 2018. [4] BERT is developed based on deep learning techniques and various methods such as semi-supervised learning, ELMo, ULMFiT, OpenAI Transformers, and Transformers. As the name implies, BERT uses Transformers. Transformer is a mechanism that studies contextual relationships between words in text. [5] Transformers can understand and convert the understanding obtained by a mechanism called the self-attention mechanism. Self-attention mechanism is Transformer's way to change the "understanding" of other related words into words that will be processed by the mechanism.

2.5. Data Validation

Evaluation of model performance using the Confusion Matrix. Confusion Matrix is a table with 4 combinations of predicted values and actual values. There are four terms in representing the results in the confusion matrix, namely, true positive, true negative, false positive, and false negative.

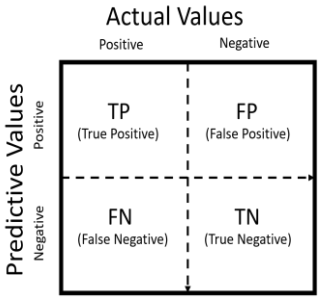

Figure 9. Confusion Matrix

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \quad (6)$$

3. RESULT AND DISCUSSION

3.1. Sentiment Analysis

Conduct sentiment analysis on all news data from 12 news portals in Indonesia. Sentiments are categorized into three types, such as: Positive, Neutral and Negative. The labelling process for sentiment is carried out by means of translating from language to English. Calculation of the polarity value to determine the sentiment on each news headline. The results of the performance evaluation of the classification using the Supervised Learning Algorithm from Machine Learning, namely the Support Vector Machine Method with the distribution of train and test data of 80:20 on 15000 total news data on all 12 news portals.

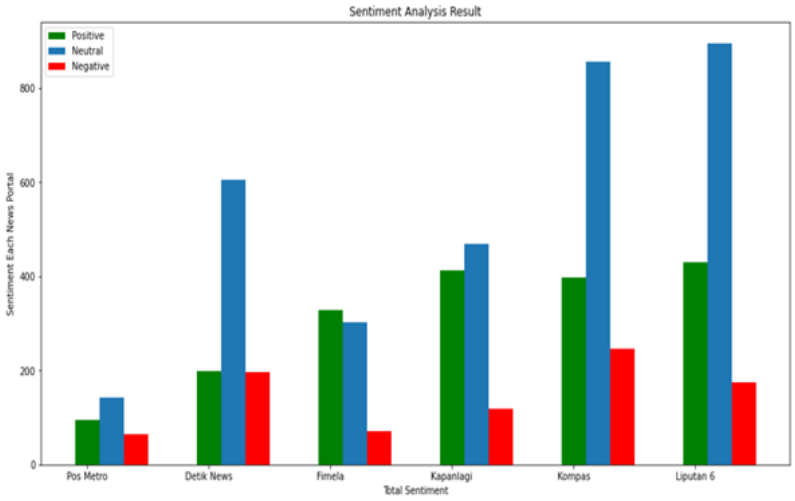

Figure 10. Sentiment Histogram on the First 6 News Portals

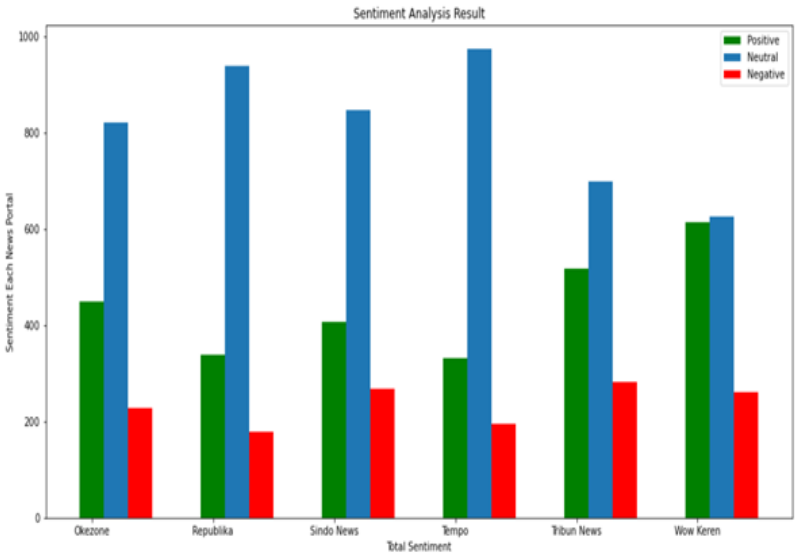

Figure 11. Sentiment Histogram on the Next 6 News Portals

Tabel 2. Sentimen Results

No	Portal	Total News	Sentiment			Percentage		
			Positive	Negative	Neutral	Positive	Negative	Neutral
1	Metro Post	300	94	64	142	31%	21%	47%
2	Detik news	1000	198	196	606	20%	20%	60%
3	Fimela	700	328	70	302	47%	10%	43%
4	When again	1000	413	118	469	41%	12%	47%
5	Compass	1500	397	246	857	26%	16%	57%
6	Coverage 6	1500	430	174	896	29%	12%	60%
7	Okayzone	1500	450	229	821	15%	30%	55%
8	Republica	1500	338	179	938	23%	12%	65%
9	Sindo News	1500	406	268	846	27%	127%	56%
10	Tempo	1500	331	194	975	13%	22%	65%
11	Tribun News	1500	518	283	699	35%	19%	47%
12	Wow, cool	1500	614	261	625	41%	17%	42%

In table 2 The Wow Keren news portal has the largest number of Positive sentiments with a total of 614 data. While the smallest number of positive sentiments is found on the Pos Metro news portal with a total of 94 data. The news portal Tribun News has the largest number of negative sentiments with 283 data. While the smallest number of negative sentiments is found on the Pos Metro news portal with 64 data. The Tempo news portal has the largest number of Neutral sentiments with a total of 975 data. While the smallest number of Neutral sentiments is found on the Pos Metro news portal with a total of 142 data. In terms of percentages for each news portal, Fimela Portal has the highest percentage of news titles with positive sentiment with a value of 47%. Meanwhile, Tempo News Portal has the smallest percentage of news titles with positive sentiment with a value of 13%. Okezone portal has the highest percentage of news with negative sentiment with a value of 30%. Meanwhile, Kapanlagi Portal, Liputan 6 and Republika have the smallest percentage of news headlines with negative sentiment with a value of 12%. The Republika and Tempo Portals have the highest percentage of news headlines with Neutral sentiment with a value of 65%. While the Wow Keren Portal has the lowest Neutral percentage with a value of 42%. On the Fimela News Portal, Kapanlagi and Wowkeren have the highest percentages of Positive and Neutral sentiment compared to other news portals. Meanwhile, Kapanlagi Portal, Liputan 6 and Republika have the smallest percentage of news headlines with negative sentiment with a value of 12%. The Republika and Tempo Portals have the highest percentage of news headlines with Neutral sentiment with a value of 65%. While the Wow Keren Portal has the lowest Neutral percentage with a value of 42%. On the Fimela News Portal, Kapanlagi and Wowkeren have the highest percentages of Positive and Neutral sentiment compared to other news portals. Meanwhile, Kapanlagi Portal, Liputan 6 and Republika have the smallest percentage of news headlines with negative sentiment with a value of 12%. The Republika and Tempo Portals have the highest percentage of news headlines with Neutral sentiment with a value of 65%. While the Wow Keren Portal has the lowest Neutral percentage with a value of 42%. On the Fimela News Portal, Kapanlagi and Wowkeren have the highest percentages of Positive and Neutral sentiment compared to other news portals.

Tabel 3. Sentiment Classification Evaluation Results

	Precision	Recall	F1-Score	Support
Negative (-)	0.78	0.65	0.71	468
Neutral (^)	0.87	0.95	0.90	1642
Positive (+)	0.89	0.81	0.85	890
accuracy			0.86	3000
macros avg	0.84	0.80	0.82	3000
weighted avg	0.86	0.86	0.86	3000

Table 3 presents the performance results of the sentiment classification task, evaluated across three sentiment classes: Negative (-), Neutral (^), and Positive (+) using standard classification metrics: precision, recall, F1 score, and support (number of samples per class). Additionally, overall metrics are reported in terms of accuracy, macro-average, and weighted average.

Tabel 4. The Performance of Clickbait Headlines Identification without Pre-processing

Features Extraction	Features Reductions	Model	Accuracy (%)	Precision (%)	Recalls (%)
TF-IDF	Non-Reduction	XG-Boost	79.60	80.55	76.98
		LightGBM	79.87	80.03	77.82
		Random Forest	80.63	80.69	78.78
		SVC	74.53	73.63	73.65
	PCC	XG-Boost	79.43	80.30	76.84
		LightGBM	79.97	80.15	77.91
		Random Forest	80.40	80.80	78.24
		SVC	75.83	75.27	73.99
	NCA	XG-Boost	79.10	79.62	76.24
		LightGBM	79.33	79.84	76.71
		Random Forest	79.80	80.10	77.05
		SVC	75.47	74.86	73.52
	relief	XG-Boost	79.87	80.16	77.84
		LightGBM	79.93	80.27	78.06
		Random Forest	80.33	80.60	78.43
		SVC	76.87	75.92	74.33
Word2Vec	Non-Reduction	XG-Boost	75.80	75.33	73.81
		LightGBM	76.13	76.32	73.52
		Random Forest	73.77	73.75	70.91
		SVC	44.77	56.97	52.40
	PCC	XG-Boost	72.17	71.48	69.75
		LightGBM	72.30	71.87	69.56
		Random Forest	70.20	69.13	68.13
		SVC	56.70	55.89	56.04
	NCA	XG-Boost	75.60	75.08	73.66
		LightGBM	75.53	75.64	72.89
		Random Forest	73.03	72.93	70.11
		SVC	57.57	52.74	51.65
	relief	XG-Boost	75.97	75.53	73.96
		LightGBM	75.80	75.84	73.25
		Random Forest	73.13	72.87	70.39
		SVC	60.20	57.49	55.97
Headline2Vec	Non-Reduction	XG-Boost	90.47	90.67	89.53
		LightGBM	90.30	90.65	89.24
		Random Forest	85.33	86.79	83.13
		SVC	93.67	93.20	93.95
	PCC	XG-Boost	88.93	89.08	87.91
		LightGBM	88.13	88.26	87.05
		Random Forest	84.27	85.23	82.21
		SVC	91.13	90.73	90.99
	NCA	XG-Boost	89.73	89.94	88.73
		LightGBM	89.23	89.54	88.11
		Random Forest	85.43	86.71	83.33
		SVC	92.93	92.54	92.93
	relief	XG-Boost	90.70	90.94	89.75
		LightGBM	89.90	90.19	88.85
		Random Forest	85.23	86.58	83.07
		SVC	94.40	94.07	94.41

Table 4 presents accuracy, precision, and recall performance metrics for various clickbait headline classification configurations without using any text preprocessing techniques. These experimental configurations included three

feature extraction methods: TF-IDF, Word2Vec, and Headline2Vec; four classifiers: XGBoost, LightGBM, Random Forest, and SVC; and four feature selection techniques: PCC, NCA, Relief, and a baseline without reduction. The most notable results were achieved by combining Headline2Vec embedding, Relief feature reduction, and a Support Vector Classifier (SVC), achieving the highest overall performance with 94.40% accuracy, 94.07% precision, and 94.41% recall. This performance significantly outperformed all other configurations, demonstrating Headline2Vec's ability to effectively capture semantic representations even without preprocessing. Furthermore, Headline2Vec's robustness was demonstrated, as its performance remained consistently high across all classifiers and reduction strategies, with accuracy exceeding 88% across all configurations. In contrast, Word2Vec embeddings showed mixed performance. The best results using Word2Vec were achieved with XGBoost without feature reduction, yielding an accuracy of 75.80%, but classification performance dropped sharply when SVC was used, dropping to 44.77% accuracy without feature reduction and even lower with PCC and NCA. This suggests that Word2Vec vectors, while effective in some boosting-based classifiers, may not be linearly separable and thus less suitable for margin-based algorithms like SVC, especially without preprocessing.

Interestingly, TF-IDF, which performed the worst in the preprocessing scenario (Table 5), showed significant improvement without preprocessing, with Random Forest achieving an accuracy of up to 80.63%. Feature reduction methods like Relief and PCC showed slight but consistent benefits when combined with TF-IDF across boosting and tree-based classifiers. However, the SVC model again performed worse than the ensemble method, suggesting that the TF-IDF feature may not be optimal for linear kernels in the presence of noise from unfiltered vocabulary. Across all feature sets, Relief and NCA consistently outperformed PCC, indicating that neighborhood- and distance-based selection strategies are better suited to capturing contextual word dependencies in raw text. PCC appeared less robust, particularly for Word2Vec and TF-IDF, which failed to maintain classification accuracy above 73%. In summary, the results in Table 4 confirm that eliminating preprocessing steps does not necessarily degrade classification performance. In fact, with the right combination of semantic embedding (Headline2Vec), feature selection (Relief), and classifier (SVC), the system achieved state-of-the-art accuracy without any preprocessing. These findings underscore the potential of embedding-driven, preprocessing-free pipelines for scalable, real-time clickbait detection in natural language processing applications.

Tabel 5. The Performance of Clickbait Headlines Identification with Pre-processing

Features Extraction	Features Reductions	Model	Accuracy (%)	Precision (%)	Recalls (%)
TF-IDF	Non-Reduction	XG-Boost	61.60	61.00	54.67
		LightGBM	61.23	59.67	54.61
		Random Forest	61.27	59.98	54.46
		SVC	59.83	62.62	50.99
	PCC	XG-Boost	59.23	29.62	50.00
		LightGBM	59.23	29.62	50.00
		Random Forest	59.23	29.62	50.00
		SVC	59.23	29.62	50.00
	NCA	XG-Boost	60.73	60.10	54.32
		LightGBM	60.40	59.83	54.01
		Random Forest	60.70	60.05	54.16
		SVC	59.43	60.30	52.27
	relief	XG-Boost	61.60	61.00	54.67
		LightGBM	61.23	59.67	54.61
		Random Forest	61.47	60.42	54.68
		SVC	59.83	62.62	50.99
Word2Vec	Non-Reduction	XG-Boost	74.03	73.35	72.05
		LightGBM	73.13	72.69	70.59
		Random Forest	70.93	70.28	68.16
		SVC	53.93	53.49	53.61
	PCC	XG-Boost	71.40	70.66	68.89
		LightGBM	71.37	70.84	68.53
		Random Forest	68.50	67.28	66.85
		SVC	60.07	60.95	51.53
	NCA	XG-Boost	73.40	72.64	71.43
		LightGBM	72.60	72.00	70.15

		Random Forest	70.37	69.67	67.50
		SVC	46.30	55.16	52.78
	relief	XG-Boost	72.90	72.10	70.89
		LightGBM	73.53	73.07	71.09
		Random Forest	70.63	69.91	67.88
		SVC	48.83	52.21	52.03
Headline2Vec	Non-Reduction	XG-Boost	90.10	90.14	89.27
		LightGBM	89.27	89.51	88.19
		Random Forest	83.00	84.04	80.76
		SVC	93.20	93.10	92.78
	PCC	XG-Boost	87.60	87.58	86.60
		LightGBM	86.73	86.87	85.50
		Random Forest	80.93	81.62	78.64
		SVC	90.47	90.61	89.58
	NCA	XG-Boost	90.10	90.14	89.27
		LightGBM	89.27	89.51	88.19
		Random Forest	83.00	84.04	80.76
		SVC	93.20	93.10	92.78
	relief	XG-Boost	90.13	90.24	89.25
		LightGBM	89.20	89.43	88.13
		Random Forest	83.00	83.79	80.90
		SVC	92.63	92.18	92.75

Table 5 presents a comparative evaluation of various combinations of feature extraction methods, feature reduction techniques, and machine learning classifiers for the clickbait headline identification task, with text preprocessing integrated as a constant across all experiments. Evaluation metrics include accuracy, precision, and recall. Among the three feature extraction techniques TF-IDF, Word2Vec, and Headline2Vec the results clearly show that Headline2Vec consistently delivers superior performance across all classifiers and reduction techniques. Specifically, the combination of Headline2Vec with Relief feature reduction and Support Vector Classifier (SVC) achieved the highest overall performance, with 92.63% accuracy, 92.18% precision, and 92.75% recall. Even without any feature reduction, Headline2Vec with SVC achieved a peak accuracy of 93.20%, highlighting the robustness of semantic embedding specifically designed for short texts.

TF-IDF demonstrated the lowest overall performance, with accuracy values remaining below 62% across all settings. Notably, applying PCC feature reduction significantly degraded classification performance across all models, with accuracy dropping uniformly to 59.23%, likely due to over-filtering or poor alignment with the textual structure of the features. Word2Vec embeddings produced moderate results, with the highest accuracy of 74.03% observed when using XGBoost without feature reduction. Feature reduction using NCA and Relief maintained comparable performance, with LightGBM achieving accuracy above 72%. However, SVC consistently performed worse with Word2Vec features, suggesting that linear separation may be inadequate for the dense vector distributions produced by these methods.

Across all feature sets, Relief and NCA outperformed PCC in retaining useful discriminatory information. This suggests that local instance-based or neighborhood-based feature reduction methods are better suited for textual classification tasks where contextual nuances are important. Overall, the results of this study confirm that semantic vectorization methods, particularly Headline2Vec, significantly improve clickbait detection performance, especially when combined with SVC or boosting algorithms and effective feature selection such as Relief. These findings highlight the importance of selecting the right combination of representation, reduction, and classification strategies to maximize accuracy in textual classification tasks.

The experimental results demonstrate that feature extraction using Headline2Vec consistently outperforms TF-IDF and Word2Vec across all classifiers. This indicates that semantic-level feature representations capture more contextual clues of clickbait patterns. Moreover, Relief as a feature reduction method shows the best synergy with SVC and XGBoost. Surprisingly, omitting the pre-processing step does not degrade performance in some cases, it even improves it suggesting that some pre-processing techniques may remove subtle indicators essential to clickbait detection. Sentiment analysis further contributes by uncovering emotional cues often embedded in clickbait, supporting classifier decisions with polarity context.

4. CONCLUSION

This study provides an in-depth comparative evaluation of clickbait headline classification under two experimental conditions: with and without conventional text preprocessing. The results show that semantic embeddings, specifically Headline2Vec, consistently outperform traditional feature extraction methods such as TF-IDF and Word2Vec, regardless of the preprocessing application. However, there are significant differences in overall system performance depending on the preprocessing implementation. In the preprocessing scenario, the best-performing configuration Headline2Vec + Relief + SVC nachieved 92.63% accuracy, 92.18% precision, and 92.75% recall. Conversely, in the no-preprocessing condition, the same configuration surpassed these performance with 94.40% accuracy, 94.07% precision, and 94.41% recall. This performance improvement suggests that omitting preprocessing can, in certain settings, preserve latent contextual signals such as informal lexical patterns or punctuation cues that might otherwise be removed during cleaning but are useful for detecting clickbait intent. Across all experiments, Headline2Vec remained the most robust feature representation, consistently maintaining over 88% accuracy across all classifiers and reduction techniques. TF-IDF, while initially weakest during preprocessing (accuracy ≤62%), improved significantly without it, reaching up to 80.63% accuracy when combined with an ensemble model like Random Forest. In contrast, Word2Vec showed more variable performance, particularly struggling with linear classifiers like SVC in both settings, highlighting the limitations of its separability in certain vector spaces. In terms of feature reduction, Relief and NCA consistently outperformed PCC, regardless of preprocessing. Their success demonstrates the effectiveness of instance-based and neighborhood-based reduction strategies in preserving semantic relationships in high-dimensional textual data. Overall, this comparative analysis emphasizes that while preprocessing is traditionally used to improve text classification, it is not always essential and, in some cases, can be detrimental to performance. These findings support the use of preprocessing-independent workflows, especially when combined with robust semantic embeddings (Headline2Vec), effective dimensionality reduction (Relief), and robust classifiers (SVC or XGBoost). This combination results in a scalable, accurate, and efficient framework for clickbait detection suitable for real-time applications, social media monitoring, and automated news validation systems..

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Universitas Negeri Surabaya, particularly the Faculty of Vocational Studies and the Department of Informatics Management, for providing the necessary facilities and support throughout the research and writing process. The authors also thank all parties who contributed to the successful completion of this study.

REFERENCES

[1] M. Sigala, A. Beer, L. Hodgson, and A. O'Connor, *Big Data for Measuring the Impact of Tourism Economic Development Programs: A Process and Quality Criteria Framework for Using Big Data*. 2019.

[2] G. Nguyen *et al.*, "Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey," Arti. Intell. Rev., vol. 52, no. 1, pp. 77–124, 2019, doi:10.1007/s10462-018-09679-z.

[3] C. Shorten and TM Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi:10.1186/s40537-019-0197-0.

[4] R. Vinayakumar, M. Alazab, KP Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi:10.109/ACCESS.2019.2895334.

[5] K. Sivaraman, RMV Krishnan, B. Sundarraj, and S. Sri Gowthem, "Network failure detection and diagnosis by analyzing syslog and SNS data: Applying big data analysis to network operations," *int. J. Innov. Technol. explore. eng.*, vol. 8, no. 9 Special Issue 3, pp. 883–887, 2019, doi:10.35940/ijitee.I3187.0789S319.

[6] AD Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors (Switzerland)*, vol. 19, no. 2, pp. 1–17, 2019, doi:10.3390/s19020326.

[7] F. Al-Turjman, H. Zahmatkesh, and L. Mostarda, "Quantifying uncertainty in internet of medical things and big-data services using intelligence and deep learning," *IEEE Access*, vol. 7, pp. 115749–115759, 2019, doi:10.1109/ACCESS.2019.2931637.

[8] S. Kumar and M. Singh, "Big data analytics for healthcare industry: Impact, applications, and tools," *Big Data Min. anal.*, vol. 2, no. 1, pp. 48–57, 2019, doi:10.26599/BDMA.2018.9020031.

[9] LM Ang, KP Seng, GK Ijemaru, and AM Zungeru, "Deployment of IoV for Smart Cities: Applications, Architecture, and Challenges," *IEEE Access*, vol. 7, pp. 6473–6492, 2019, doi:10.109/ACCESS.2018.2887076.

[10] BPL Lau *et al.*, "A survey of data fusion in smart city applications," Inf. Fusion, vol. 52, no. January, pp. 357–374, 2019, doi:10.1016/j.inffus.2019.05.004.

[11] Y. wu *et al.*, "Large scale incremental learning," Proc. IEEE Computing. soc. conf. Comput. vis.

Pattern Recognit., vol. 2019-June, pp. 374–382, 2019, doi:10,1109/CVPR.2019.00046.
[12] A. Mosavi, S. Shamshirband, E. Salwana, K. wing Chau, and JHM Tah, "Prediction of multi-inputs bubble column reactor using a novel hybrid model of computational fluid dynamics and machine learning," *eng. app. Comput. Fluid Mechs.*, vol. 13, no. 1, pp. 482–492, 2019, doi:10.1080/19942060.2019.1613448.
[13] V. Palanisamy and R. Thirunavukarasu, "Implications of big data analytics in developing healthcare frameworks – A review," *J. King Saud Univ. - Computing. inf. science.*, vol. 31, no. 4, pp. 415–425, 2019, doi:10.1016/j.jksuci.2017.12.007.
[14] J. Sadowski, "When data is capital: Datafication, accumulation, and extraction," *Big Data Soc.*, vol. 6, no. 1, pp. 1–12, 2019, doi:10.1177/2053951718820549.
[15] JR Saura, BR Herraez, and A. Reyes-Menendez, "Comparing a traditional approach for financial brand communication analysis with a big data analytics technique," *IEEE Access*, vol. 7, pp. 37100–37108, 2019, doi:10.109/ACCESS.2019.2905301.
[16] D. Nallaperuma *et al.*, "Online Incremental Machine Learning Platform for Big Data-Driven Smart Traffic Management," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 12, pp. 4679–4690, 2019, doi:10.109/TITS.2019.2924883.
[17] S. Schulz, M. Becker, MR Groseclose, S. Schadt, and C. Hopf, "Advanced MALDI mass spectrometry imaging in pharmaceutical research and drug development," *Curr. Opinion. Biotechnol.*, vol. 55, pp. 51–59, 2019, doi:10.1016/j.copbio.2018.08.003.
[18] C. Shang and F. You, "Data Analytics and Machine Learning for Smart Process Manufacturing: Recent Advances and Perspectives in the Big Data Era," *Engineering*, vol. 5, no. 6, pp. 1010–1016, 2019, doi:10.1016/j.eng.2019.01.019.
[19] Y. Yu, M. Li, L. Liu, Y. Li, and J. Wang, "Clinical big data and deep learning: Applications, challenges, and future outlooks," *Big Data Min. anal.*, vol. 2, no. 4, pp. 288–305, 2019, doi:10.26599/BDMA.2019.9020007.
[20] M. Huang, W. Liu, T. Wang, H. Song, X. Li, and A. Liu, "A queuing delay utilization scheme for on-path service aggregation in services-oriented computing networks," *IEEE Access*, vol. 7, pp. 23816–23833, 2019, doi:10.1109/ACCESS.2019.2899402.
[21] G. Xu, Y. Shi, X. Sun, and W. Shen, "Internet of things in marine environment monitoring: A review," *Sensors (Switzerland)*, vol. 19, no. 7, pp. 1–21, 2019, doi:10.3390/s19071711.
[22] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and SM Altowaijri, *Smarter traffic prediction using big data, in-memory computing, deep learning and gpus*, vol. 19, no. 9. 2019.
[23] S. Leonelli and N. Tempini, *Data Journeys in the Sciences*. 2020.
[24] N. Stylos and J. Zwiegelaar, *Big Data as a Game Changer: How Does It Shape Business Intelligence Within a Tourism and Hospitality Industry Context?* 2019.
[25] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *arXiv*, vol. 13, no. 1, 2017.

AUTHOR BIOGRAPHY

	**Anisa Nur Azizah** is a lecturer in the Informatics Engineering, Faculty of Engineering, Universitas Wijaya Putra, Indonesia. She earned her Bachelor of Mathematics (S.Mat.) in Mathematics from Sunan Ampel State Islamic University, Surabaya, in 2021. She later obtained her Master of Computer Science (M.Kom.) in Informatics Engineering from Institut Teknologi Sepuluh Nopember, Surabaya, in 2023. Her primary research interests include Artificial Intelligence and Image Processing. She can be contacted via email at: anisanurazizah@uwp.ac.id.
	**Misbachul Falach Asy'ari** is a researcher in the Department of Informatics Engineering at Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. He earned both his Bachelor's degree (S.Kom.) and Master's degree (M.Kom.) in Informatics Engineering from Institut Teknologi Sepuluh Nopember, completing his postgraduate studies in 2023. His primary research interests include Machine Learning, Data Mining and Image Processing. He can be contacted via email at: misbachulfalach@gmail.com.
	**Moch Deny Pratama** is a lecturer in the Department of Informatics Management at the Universitas Negeri Surabaya, Indonesia. He earned his Bachelor of Applied Computer Science (S.Tr.Kom.) in Informatics Engineering from the State Polytechnic of Malang in 2021. He later obtained his Master of Computer Science (M.Kom.) in Informatics Engineering from Institut Teknologi Sepuluh Nopember, Surabaya, in 2023. His primary research interests include Data Science, Machine Learning, Algorithm Design, and Artificial Intelligence. He can be contacted via email at: mochpratama@unesa.ac.id.
	**Dimas Novian Aditia Syahputra**is a lecturer at the Department of Informatics Management, Universitas Negeri Surabaya, Indonesia. He earned his Bachelor of Applied Electrical Engineering (S.Tr.T) from the Surabaya State Electronics Polytechnic in the field of Electrical Engineering, Surabaya in 2019. He also earned his Master of Applied Engineering (M.Tr.T.) from the Surabaya State Electronics Polytechnic in the field of Electrical Engineering, Surabaya in 2022. He mainly researches the Internet of Things and AR/VR. He can be contacted via email: dimassyahputra@unesa.ac.id.
	**M Adamu Islam Mashuri** is a lecturer at the Department of Informatics Management, Surabaya State University, Indonesia. He earned his Bachelor of Applied Engineering (S.Tr.T) from the Surabaya State Electronics Polytechnic in Telecommunication Engineering, Surabaya in 2021. He also earned his Master of Applied Computer (M.Tr.Kom) from the Surabaya State Electronics Polytechnic in Informatics and Computer Engineering, Surabaya in 2023. He mainly researches the Internet of Things and Artificial Intelligence. He can be contacted via email: mmashuri@unesa.ac.id.
	**Binti Kholifah** is a lecturer in the Department of Informatics Management, Surabaya State University, Indonesia. She earned her Bachelor of Computer Science (S.Kom) from Maulana Malik Ibrahim State Islamic University in Informatics Engineering, Malang in 2019. She also earned her Master of Applied Computer Science (M.Tr.Kom) from Surabaya State Electronics Polytechnic in Informatics and Computer Engineering, Surabaya in 2021. She primarily research Game Programming and Mobile Programming. She can be contacted via email: bintikholifah@unesa.ac.id.

	**Rifqi Abdillah** is a lecturer in the Department of Informatics Engineering, Surabaya State University, Indonesia. He earned his Bachelor of Applied Engineering (S.Tr.T) from the Electronic Engineering Polytechnic Institute of Surabaya in 2021, majoring in Informatics Engineering. He subsequently earned his Master of Computer Science (M.Kom) from the Institut Teknologi Sepuluh Nopember, Surabaya in 2023. His primary research include Machine Learning, Applied Artificial Intelligence, and Internet of Things. He can be contacted via email: rifqiabdillah@unesa.ac.id.
	**Adinda Putri Pratiwi** is a researcher in the Department of Informatics Engineering, Institut Teknologi Sepuluh Nopember, Indonesia. Her majoring in Informatics Engineering subsequently earned her Master of Computer Science (M.Kom) from the Institut Teknologi Sepuluh Nopember, Surabaya in 2023, focusing on Data Mining and Intelligent Systems. Her primary research interests include Machine Learning, Data Mining, and Internet of Things. Her can be contacted via email: adindaputripratiwi97@gmail.com.
	**Dina Zatusiva Haq** is a lecturer in the Informatics, Faculty of Computer Science, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia. She earned her Bachelor of Mathematics (S.Mat.) in Mathematics from Sunan Ampel State Islamic University, Surabaya, in 2021. She later obtained her Master of Computer Science (M.Kom) in Informatics Engineering from Institut Teknologi Sepuluh Nopember, Surabaya, in 2023. Her primary research interests include Artificial Intelligence. She can be contacted via email at: dinaza.if@upnjatim.ac.id.