Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2025. Vol. 4. No. 2; e-ISSN: 2808-4519

Optimizing Email Spam Classification Using Naïve Bayes and Principal Component Analysis

Shinta Virgiana[1]*, Rudi Kurniawan[2], Tati Suprapti[3]

[1,2,3]STMIK IKMI CIREBON
Jl. Perjuangan No. 10B Majasem, Kesambi, Cirebon, Indonesia
shintavirgin24@gmail.com [1]*

Abstract

In the ever-evolving digital era, email spam filtering is an important challenge to maintain the security and comfort of email services. The Naïve Bayes algorithm is widely used for spam email classification because of its ability to manage large data, although there are still limitations in terms of accuracy, precision and recall. This research aims to improve spam email classification performance by combining Naïve Bayes and Principal Component Analysis (PCA) to optimize model accuracy and explore optimal parameters in the reduction dimension. The research methodology goes through the Knowledge Discovery in Database (KDD) stages which include selection, preprocessing, transformation using PCA, development of a classification model using Naïve Bayes, and evaluation of model performance. The dataset used consists of emails categorized as spam and non-spam. The experimental results show that the combination of Naïve Bayes and PCA achieves the highest accuracy of 99.24% with 7 principal components. The fixed number of components approach shows better performance compared to preserving variance, emphasizing the importance of selecting appropriate PCA parameters in improving the effectiveness of model classification. This research shows that PCA not only reduces the complexity of the dataset but also increases the efficiency of the classification algorithm.

***Keywords**: spam classification, knowledge discovery in database (KDD), naïve bayes, principal component analysis (PCA), dimensionality reduction*

1. Introduction

The use of the internet has become an important necessity in communication life, with email as one of the main facilities that is often used. Email is not only used to exchange personal messages, but is also an important tool in the world of business, education and government [1]. Data shows that by 2023, approximately 347 billion emails are sent and received every day globally, reflecting the important role email plays in modern life [2]. However, the popularity of email also presents new challenges, one of which is the existence of spam email. Spam emails are unwanted messages and can interfere with communications, and potentially threaten the security of user data. To overcome this problem, email spam classification systems have an important role in filtering incoming emails so that users can avoid unwanted messages. Even though many methods have been developed, the level of accuracy and efficiency in spam detection is still a big challenge [3]. One method that is often used is the Naïve Bayes algorithm, because of its simplicity and effectiveness in processing text for classification.

However, this algorithm has limitations, especially when faced with data with high features or great complexity. The main challenge in email spam classification with Naïve Bayes is dealing with the problem of redundant features or high-dimensional data, and ensuring a high level of accuracy, precision and sensitivity, while optimizing the efficiency of the classification process [4]. One solution that can be applied is to use Principal Component Analysis (PCA), which aims to extract the main features from an existing data set, so that the data becomes simpler and can be processed more efficiently [5]. Previous research has shown that the combination of the Naïve Bayes algorithm with dimensionality reduction techniques such as PCA can improve model performance, especially in terms of accuracy and processing time [6]. Thus, it becomes important to determine optimal dimensionality reduction parameters, including the number of principal components, to improve the performance of spam classification models.

2. Research Methods

This research applies quantitative methods with an experimental approach, where the spam email dataset is analyzed numerically to produce an optimal classification model. The main focus of the research is to improve the accuracy of email spam classification models through a combination of Naïve Bayes and Principal Component Analysis (PCA) algorithms. The research stages refer to the Knowledge Discovery in Database (KDD) approach, starting from data collection, selecting relevant data, pre-processing to ensure data quality, to data transformation so that it is ready for further analysis. After the data has been transformed, a data mining process is carried out by applying a combination of Naïve Bayes and PCA algorithms to build an email spam classification model. The final stage is evaluation, where the analysis results are evaluated to measure the effectiveness of the resulting model, with an emphasis on increasing classification accuracy. The research method used in this research is as shown in Figure 1.

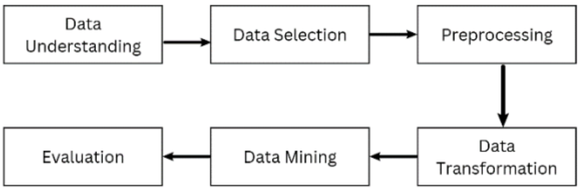

Figure 1: Stages of research methods

2.1. Data Collection Techniques

The data collection technique in this research uses observation techniques. Research data was obtained through secondary data collection techniques from trusted public repositories. Data collection was carried out by accessing the public repository, namely www.kaggle.com and downloading the dataset in ZIP format which was then extracted into CSV format. The dataset obtained was then confirmed to be relevant to the research criteria before further analysis was carried out.

2.2. Data Analysis Technique

This research uses a systematic Knowledge Discovery in Database (KDD) framework, including selection, preprocessing, data transformation, data mining, and evaluation. Each stage is designed to ensure optimal data processing, identification of relevant patterns, and development of accurate classification models. This KDD approach aims to optimize classification models, in particular improving the performance of the Naïve Bayes algorithm combined with PCA, as well as contributing to the development of classification techniques for large and complex data. This systematic flow is important to ensure the data used is relevant, of high quality, and supports the development of accurate models, while enabling more efficient data processing, minimizing errors, and maximizing efficiency on large and diverse datasets, thereby producing measurable and accurate results for decision-based scientific and objective data. The KDD process, as illustrated in Figure 2, follows the stages commonly described in previous literature [7].

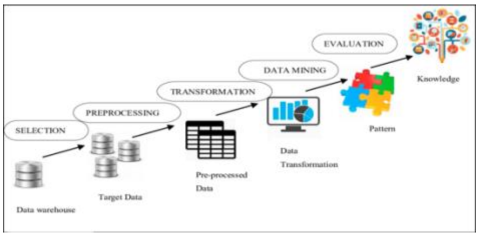

Figure 2: KDD Flow

3. Result and Discussion

3.1. Result

The results and discussion of this research describe efforts to improve spam email classification models through combining the Naive Bayes algorithm and Principal Component Analysis (PCA). This research aims to improve the accuracy of the spam classification system by utilizing PCA to reduce high data dimensions, so as to optimize the performance of the Naive Bayes algorithm. The results of the combination of these two methods were evaluated to identify the contribution of PCA in increasing the efficiency and accuracy of the model. The discussion also involves comparative analysis between the Naive Bayes model equipped with PCA and the Naive Bayes model without PCA, as well as evaluating classification performance based on matrices such as accuracy, precision and recall. It is hoped that the findings from this research will provide new insights in developing more accurate and efficient spam email classification models.

3.1.1. Data Understanding

This research uses a spam email dataset obtained from Kaggle in CSV format. This dataset was chosen due to the availability of diverse and relevant information for spam/non-spam email classification, as well as its consistent quality and frequent use in similar studies, allowing comparison of results. The initial process includes downloading and processing the data to ensure the format is suitable for the analysis tools, as well as identifying the structure, content and quality of the data. The dataset used consists of 4,601 records with 58 numeric attributes, representing email characteristics such as word frequency *(word_freq),* symbols *(char_freq),* and capital letter patterns *(capital_run_length).* Processing this dataset is crucial for the development and evaluation of the developed classification model. The Kaggle dataset was chosen because of its consistent quality and is often used in various similar studies, so that the findings can be compared with previous research. Apart from that, the clean condition of the dataset makes it easier to develop a classification model. This can be seen from the results of the statistical analysis shown in Table 1 below.

Table 1: Statistic Data

Name	Type	Missing
Spam	Nominal	0
Prediction(Spam)	Nominal	0
Confidence(1)	Real	0
Confidence(0)	Real	0

Word_freq_make	Real	0
…	…	…
Capital_run_length_longest	Integer	0

Table 1 shows that the dataset has gone through good preprocessing, with each attribute clean, free from missing values and inconsistencies. The absence of a value in the "Missing" column confirms this. This standardized dataset provides a strong foundation for subsequent analysis, allowing research to focus on applying analytical models for accurate and effective results. In short, the dataset is clean and ready for analysis.

3.1.2. Data Selection

The data selection stage uses a relevant and high quality CSV format dataset, crucial for research into the accuracy of email spam classification. To read a CSV dataset in Altair Studio, use the "Read CSV" operator Figure 3 as the initial step in the analysis flow. This operator integrates raw data directly from CSV files, minimizing the potential for manual conversion errors, and increasing analysis efficiency and accuracy.

Figure 3: Operator Read CSV

After the data is loaded using the Read CSV operator, the data selection stage continues with the application of the Set Role operator as shown in Figure 4

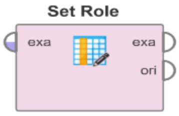

Figure 4: Operator Set Role

The Set Role operator in Figure 3 functions to define the role of each attribute, such as labels (targets) and features (predictors). In this research, email classification attributes (spam/non-spam) are defined as labels, while other attributes are as features. The use of these operators ensures that the data is properly prepared for optimal processing by the analysis algorithm. The parameters used in the set role operator are shown in Table 2.

Tabel 2: Set Role Parameters

Parameters	**Value**
Attribute Name	Spam
Target Role	Label

The results of using the Operator Set Role appear in Table 3 with 57 regular attributes, 1 special attribute and 4,601 records.

Table 3: of results from using the set role operator

Spam	**Word_freq_make**	**…**	**Capital_run_length_total**
1	0	…	278
…	…	…	…
0	0	…	40

3.1.3. Preprocessing

After the data selection stage, the dataset used in this research has met the quality and relevance criteria required for further analysis. The data was proven to be clean and complete, as shown in Table.1, thereby eliminating the need for additional preprocessing procedures such as handling missing values, removing duplicate data, or attribute normalization. The "Read CSV" operator has been implemented to read the dataset, and the "Set Role" operator has been applied to define the role of each attribute in the dataset. With the dataset ready for analysis, the research focus shifted to constructing a classification model with the main aim of optimizing accuracy in distinguishing between spam and non-spam emails. This condition facilitates a more efficient analysis process and ensures that the main attention is allocated to optimal evaluation and improvement of model accuracy.

3.14. Data Transformation

To increase analytical flexibility in developing models combining the Naive Bayes algorithm with PCA, data transformation was carried out to convert numerical attributes into categorical (polynomial) representations. To facilitate grouping numeric values into categories in a structured manner, the Numerical to Polynomial operator is implemented, as illustrated in Figure 5.

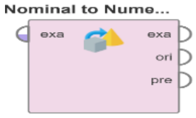

Figure 5: Numerical To Polynominal Operator

The parameters used in the Numerical to polynominal operator are shown in Table 4.

Table 4: numerical to polynominal parameters

Parameters	**Value**
Attribute filter type	Subset

Attributes	Select attributes #spam

The results of using the numerical to polynominal operator appear in Table 5.

Table 5: the result of using the numerical to polynominal operator

Name	**Type**	**Transformation**	
		Before	**After**
Spam	Nominal	0	1
		1	0

Table 4 illustrates the use of attribute parameters in the Select Attributes operator which is focused on selecting the #spam attribute for the transformation process. This transformation changes the data type of the #spam attribute from Integer to Nominal. This method allows efficient data transformation without affecting other numeric attributes that remain necessary in the original format. This operator provides flexibility in customizing the resulting categories according to analysis needs. This data transformation is a crucial step to ensure compatibility of the data format with the model development in this research. From the stage of selecting data to data transformation, the series of operators used is shown in Figure 6.

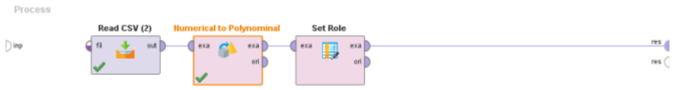

Figure 6: The process of selecting data to data transformation

Figure 5 represents the operators used from the data selection stage to data transformation. In implementing the KDD process, the order of operator use is flexible and can be adjusted to technical needs and the characteristics of the dataset being processed. In Figure 5, the implementation of the Numerical to Polynomial operator comes before the Set Role operator. This strategy aims to facilitate the transformation of numerical attributes into categorical (polynomial) representation and ensure consistency of the numerical attribute format, so that the next stages can be executed efficiently and purposefully. This sequence adjustment indicates that the operator usage flow does not always follow the KDD stages linearly, but is iterative or adaptive, depending on the analysis needs.

3.1.5. Data Mining

After the dataset has gone through the data selection and transformation stages and is confirmed to be clean and ready to be analyzed, the next stage in the Knowledge Discovery in Database (KDD) process is data mining. At this stage, the data is trained using a combination of the Naïve Bayes Algorithm model with PCA for spam email classification. One of the crucial aspects in implementing PCA is evaluating the covariance between attributes. For this purpose, the Covariance Matrix Operator is implemented, as illustrated in Figure 7.

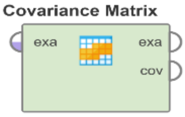

Figure 7. Covariance Matrix Operator

The Covariance Matrix operator operates automatically without requiring additional parameter configuration. Covariance calculations are carried out directly based on the attributes contained in the dataset, thus facilitating efficient and accurate analysis of linear relationships between variables. This operator applies the trained model to the dataset without requiring further adjustments. Through the Covariance Matrix Operator, evaluation is carried out by computing the covariance between attributes, which is a statistical measure to describe the extent to which two attributes change simultaneously. The results of covariance calculations between attributes are shown in Table 6.

Table 6: Pairwise covariance matrix

First Attribute	**Second Attribute**	**Covariance**
Word_freq_make	Word_freq_orders	-0.007
…	…	…
Capital_run_length_longest	Capital_run_length_total	56189.02

Based on the results of the covariance analysis obtained through the Covariance Matrix Operator, the next stage is the application of Principal Component Analysis (PCA) to reduce the dimensions of the dataset. The Principal Component Analysis operator implemented in Figure 8 functions to reduce dimensions while maintaining important information in the data.

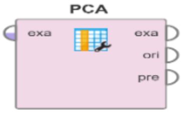

Figure 8: Principal component analysis operator

The parameters used in this operator are presented in the Table 7.

Table 7. Primcipal component analysis parameters

Parameters	Value	
Dimensionality Reduction	Keep Variance	Fixed Number
Variance Threshold	0,10 – 1.0	
Number of component		1 - 57

The results of using the Principal Component Analysis Operator using number of components 7 are shown in Table 8.

Table 8: Result of PCA operator

Row No.	Spam	Pc 1	…	Pc 7
1	1	-3.787	…	0.113
…	…	…	…	…
4601	0	-247.755	…	0.021

Table 8, shows the new dataset with lower dimensions. This reduced dataset is then used as input for the classification model in the "With PCA" subprocess. This dimension reduction is expected to increase classification accuracy and mitigate the risk of overfitting, especially on complex datasets. The results of the "With PCA" subprocess will be compared with the "Without PCA" subprocess to evaluate the effectiveness of PCA in improving model performance. The implementation of the two subprocesses is illustrated in Figure 9.

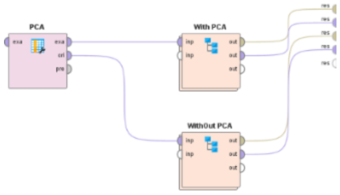

Figure 9: Principal component analysis operator

Figure 9, illustrates the connection of the Principal Component Analysis Operator with two subprocesses, namely "With PCA" and "Without PCA," through separate processing flows. The "With PCA" subprocess receives input from the exa path (example set), which contains the PCA reduced dataset, for further processing. This dimensionality reduction aims to increase computational efficiency and mitigate the risk of overfitting, especially on complex datasets with many attributes. This reduced dataset is then processed using a classification algorithm. In contrast, the “No PCA” subprocess receives input from the original path, which contains the original dataset without dimensionality reduction, for comparison of model performance between the reduced and original datasets. After dimension reduction, the dataset is prepared for training and testing using the Split Data operator, the Naïve Bayes classification algorithm, as well as model testing and evaluation, all of which are in the PCA operator as shown in Figure 10.

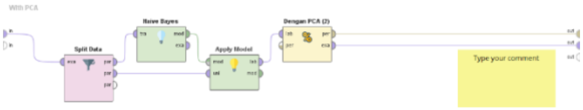

Figure 10: Process inside the PCA operator

The dimensionally reduced dataset is then partitioned into two subsets: training data and testing data for objective evaluation of the classification model on data that has never been seen (testing data). This division is implemented using the Split Data Operator with the proportions as in Table 9.

Table 9: Split Data parameters

Parameters	Value
Partition	Edit enumeration 0,8 : 0,2
Sampling type	Linear sampling

Based on table 9, 80% of the data is allocated for training and 20% for testing. The linear sampling method is used to ensure an orderly distribution of data without changing the original structure. The results of this division produce 3681 training data instances and 920 testing data instances out of a total of 4601 instances, as in Table 10.

Table 10: Result of split data

Row No	Spam	PC 1	…	PC 7
1	0	-284.911	…	0.131
…	…	…	…	…
920	0	-247.755	…	0,021

Next, a subset of the training data is processed using the Naïve Bayes algorithm, a probabilistic-based algorithm that utilizes Bayes' Theorem with the assumption of independence between attributes. The Laplace correction parameter is activated to overcome the zero probability problem as shown in Table 11.

Table 11: Naïve Bayes Parameters

Parameters	Value
Laplace correction	☑

The results of using naive Bayes are as follows
SimpleDistribution
Distribution model for label attribute spam
Class 1 (0.493)
57 *distributions*
Class 2 (0.507)
57 *distributions*

After training the model, the Apply Model operator is applied to predict classes on the test data, connecting the training and prediction stages for model performance evaluation. As shown in Table 12.

Table 12: Model testing result

Row No	Spam	Predictions (spam)	Confidence(1)	Confidence(2)	…	Capital_run_length_total
1	0	0	0.000	1.000	…	3
…	…	…			…	…
920	0	1	0.967	0.003	…	40

Next, the Performance Operator is used to evaluate model performance on data with and without PCA, comparing evaluation metrics such as accuracy, precision, and recall. The parameters used in this operator are shown in Table 13.

Table 13: Performance Parameters

Parameters	Value
Main criterion	First
Accuracy	☑
Skip Undefined Label	☑
Use example weights	☑
Class weight	Edit list : Weights = 1.0

The results of using operator performance on the confusion matrix with PCA and without PCA are shown in Table 14 and Table 15.

Table 14: PerformanceVector on confusion matrix with PCA

Accuration : 99,24%			
	True 1	True 0	Class precission
Pred 1	0	7	0,00%
Pred 0	0	913	100%
Class recall	0,00%	99,24%	

Table 15. PerformanceVector on confusion matrix without PCA

Accuration : 58,48%			
	True 1	True 0	Class precission
Pred 1	0	382	0,00%
Pred 0	0	538	100%
Class recall	0,00%	58,48%	

3.1.6. Evaluation

After the data mining stage, model performance evaluation was carried out to measure classification accuracy on test data and compare the effectiveness of using PCA, including evaluating two dimension reduction approaches, namely fixed number and keep variance. These two parameters have different implications on principal component selection and model performance. This evaluation aims to determine the most suitable parameters for implementing PCA with optimal accuracy.

a. Fixed number

The fixed number approach to PCA allows direct determination of the number of main components to be retained. The research dataset has 58 attributes, so a comparative analysis of the number of components was carried out. The analysis results show that a fixed number with 7 components produces optimal accuracy of 99.24% Table 16.

Table 16: Optimal accuracy value for fixed numbers

Fixed Number	
Number Of components	Keep Variance
1	97,72%
2	98,37%
3	99,02%
4	98,59%
5	98,49%
6	98,26%
7	99,24%
…	…
57	58,48%

Table 16 indicates that these 7 components represent most of the data variance and retain significant information. This approach increases computational efficiency and maintains high classification performance, in contrast to analysis without PCA which shows a significant decrease in accuracy of up to 58.48% as shown in Table 17.

Table 17: Comparison of accuracy values with and without PCA using fixed number parameters

	Dengan PCA	**Tanpa PCA**
Accuration	99,24%	58,48%

b. Keep Variance

In the keep variance approach, component selection is carried out adaptively based on their contribution to the total variance of the data. The results of the keep variance parameter analysis show optimal accuracy in Table 18.

Table 18: Optimal accuracy value for keep variance

Keep Variance	
Variance Threshold	Accuration
0,10	97,72%
0,15	97,72%
…	…
0,95	98,37%
1,0	54,67%

The accuracy value is relatively stable at 97.72%, with the highest accuracy being 98.37% at a variance threshold of 0.95, which is still lower than the optimal fixed number result. Accuracy drops drastically to 54.67% when the entire variance (1.0) is maintained. By maintaining the optimal variance proportion, PCA significantly improves model accuracy compared to analysis without dimension reduction (58.48%), which is often inefficient and less than optimal in utilizing dataset information. Comparison of accuracy with and without PCA using keep variance is shown in Table 19.

Table 19: Comparison of accuracy values with and without PCA using keep variance parameters

	Dengan PCA	**Tanpa PCA**
Accuration	98,37%	58,48%

Based on the analysis of the two parameters, fixed number is proven to have the highest accuracy compared to keep variance. Fixed number allows direct determination of the optimal number of principal components, achieving a balance between computational efficiency and classification accuracy. Even though the keep variance is stable for most variance thresholds, it is less effective in producing optimal accuracy. It can be concluded that fixed number with 7 components is the optimal dimension reduction method in this classification model.

Evaluation of the classification model without dimension reduction in Table 15 shows low accuracy, namely 58.48%, far below the accuracy achieved with dimension reduction using PCA, especially with fixed number parameters which reached 99.24%. This low accuracy indicates that an excessive number of attributes can make it difficult for the model to identify relevant patterns, reducing classification performance. Uncontrolled dataset complexity also increases the risk of overfitting, which hinders model generalization on test data. These findings emphasize the importance of dimensionality reduction for the efficiency and accuracy of classification models, especially on datasets with many and complex attributes. Comparison of accuracy with and without dimension reduction is shown in Table 17.

Apart from accuracy, model performance is also evaluated based on precision and recall metrics. Since these two metrics tend to be stable across both methods (with and without dimensionality reduction), further analysis is needed to understand the contribution of each metric to the overall model performance assessment.

3.1.2. Discussion

Analysis of the performance of the classification model shows very good results, with accuracy reaching 99.24% after applying dimensionality reduction using PCA. Increased accuracy of 40.76% compared to a model that only uses Naïve Bayes without dimension reduction. The accuracy values in Table 14 can be calculated manually using the following equation.

$$Accuration = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

$$Accuration = \frac{0+913}{0+913+0+7} = \frac{913}{920} = 0{,}99239 \sim 0{,}9924$$

$$0{,}99239 \sim 0{,}9924\ = 99{,}24\%$$

Evaluation of the dimensionality reduction parameter was also carried out by showing that the fixed number produced an optimal accuracy of 99.24%, exceeding the keep variance which was only 98.37% even though the difference was relatively small, only 0.87%. Therefore, a fixed number with 7 components was chosen as the optimal dimension reduction method. The selection of 7 components was proven to be effective in representing most of the variance in the dataset.

4. Conclusion

After conducting research by developing a combination of the Naïve Bayes algorithm model and principal component analysis, the following conclusions were obtained.

1. Based on the results of the research carried out, the classification model achieved the best accuracy value of 99.24%, which shows excellent overall model performance. In the precision value, the model succeeded in getting a value of 100% for the negative class, and the precision value for the positive class was 0%. Meanwhile, the recall value in the model reached 99.24% for the negative class, and for the positive class the recall was 0. These results show that the model has very good performance in detecting non-spam emails (negative class).
2. The dimensionality reduction parameter with the best accuracy is obtained using the fixed number approach which gives the highest accuracy results, compared to other methods, with a comparison of accuracy reaching 99.24% when using PCA compared to the model without PCA which only achieves an accuracy value of 58.48%. This shows that the model developed is effective and successful, because it experienced a significant increase in accuracy, namely 40.76%. Thus, this approach can be further applied to optimize classification models on various other types of text data, especially in the case of content filtering such as email phishing detection or sentiment classification, where high accuracy is needed to increase user accessibility.
3. The optimal PCA value to obtain the best accuracy is obtained with a number of components of 7, which shows that data dimension reduction at this level is able to maintain relevant information without losing model performance. These results confirm that the Principal Component Analysis approach in increasing data efficiency can support the performance of the Naïve Bayes algorithm, especially in email spam classification, so that the resulting model is more reliable for filtering emails accurately and effectively.

5. Reference

[1] I. AbdulNabi and Q. Yaseen, “Spam email detection using deep learning techniques,” *Procedia Comput. Sci.*, vol. 184, no. 2019, pp. 853–858, 2021, doi: 10.1016/j.procs.2021.03.107.

[2] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, “A comprehensive survey for intelligent spam email detection,” *IEEE*

Access, vol. 7, pp. 168261–168295, 2019, doi: 10.1109/ACCESS.2019.2954791.

[3] D. A. Anggraini, M. Ikhsan, and S. Suhardi, "Implementation of the Naïve Bayes Algorithm in the SMS Spam Filtering System," *J. Comput. Networks, Archit. High Perform. Comput.*, vol. 6, no. 2, pp. 838–849, 2024, doi: 10.47709/cnahpc.v6i2.3875.

[4] M. Anita, B. Susanto, and L. Larwuy, "Perbandingan Metode Random Forest dan Naïve Bayes dalam Email Spam Filtering," *KUBIK J. Publ. Ilm. Mat.*, vol. 7, no. 2, pp. 88–96, 2023, doi: 10.15575/kubik.v7i2.18933.

[5] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, 2019, doi: 10.1016/j.heliyon.2019.e01802.

[6] H. Mukhtar, J. Al Amien, and M. A. Rucyat, "Filtering Spam Email menggunakan Algoritma Naïve Bayes," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 1, pp. 9–19, 2022, doi: 10.37859/coscitech.v3i1.3652.

[7] M. D. Akbar, M. Martanto, and ..., "Klasifikasi Motif Batik Jawa Menggunakan Algoritma K-Nearest Neighbors (Knn)," *JURSIMA (Jurnal ...*, vol. 10, no. 2, 2022, [Online]. Available: https://ejournal.indobarunasional.ac.id/index.php/jursima/article/view/412%0Ahttps://ejournal.indobarunasional.ac.id/index.php/jursima/article/download/412/275