



Performance Analysis Algorithm Classification and Regression Trees and Naive Bayes Based Particle Swarm Optimization for Credit Card Transaction Fraud Detection

Rita Afridah*, Munirul Ula, Lidya Rosnita

Department of Informatics, Universitas Malikussaleh, Aceh, Indonesia

*Corresponding author Email: rita.200170140@mhs.unimal.ac.id

The manuscript was received on 20 January 2024, revised on 4 March 2024, and accepted on 26 June 2024, date of publication 25 August 2024

Abstract

With the advancement of technology, credit cards have become a popular tool for transactions, both physically and online, due to their ease of use and seamless integration with banking systems. However, with the increasing use of credit cards, the cases of fraud have also risen, resulting in financial losses for both cardholders and banks. To address this issue, effective and efficient credit card transaction fraud detection has become a top priority. Using machine learning algorithms is one of the techniques that can be employed to detect fraud in credit card transactions. The purpose of this research is to determine the performance and find the best method of the CART algorithm, Naive Bayes, and their combination with Particle Swarm Optimization (PSO) in detecting fraud in credit card transaction histories. The data used consists of 568,630 big data entries with parameters including id, V1-V28, amount, and class. The research results obtained are as follows: the accuracy of the Naive Bayes algorithm is 93.15%, precision is 94%, recall is 93%, and AUC is 0.99. For the CART algorithm, the accuracy is 99.96%, with precision and recall at 100%, and AUC at 1.00. Additionally, the Naive Bayes algorithm combined with PSO achieved an accuracy of 98.50%, precision and recall of 98%, and AUC of 1.00. Lastly, the CART algorithm combined with PSO reached an accuracy of 99.97%, with precision and recall at 100%, and AUC at 1.00. It can be concluded that the best method resulting from the tests conducted is the Classification and Regression Trees method combined with Particle Swarm Optimization.

Keywords: Classification and Regression Trees, Fraud, Machine Learning, Naive Bayes, Particle Swarm Optimization.

1. Introduction

In today's digital era, which is advancing rapidly and significantly, the use of credit cards has seen a substantial increase due to their ease of use and seamless integration with banking systems. A credit card is one of the payment methods that replaces cash for transactions, provided by financial institutions such as banks. A credit card offers a facility in the form of credit to its holder, where at the due date, the owner can pay a minimum amount specified, and the remaining balance is treated as credit [1], [2]. However, along with the increase in credit card usage, cases of crime are also vulnerable to occur, such as cybercrime. The crime that is susceptible to happen during credit card transactions is fraud.

Fraud is a deception in credit card transactions that can harm both customers and banks or companies, carried out for the purpose of personal gain [3]. In banking operations, according to Chapter 1, Article 1, of the Financial Services Authority Regulation of the Republic of Indonesia Number 39/POJK 03/2019 concerning the Implementation of Anti-Fraud Strategies for Commercial Banks, it states that fraud is a deviation or intentional negligence by an individual against the bank, its customers, or other parties to deceive, mislead, or manipulate anything that occurs within the bank's environment or while using banking facilities, resulting in losses for the bank, its customers, or other parties. The issue of fraud continues to increase every year and has become a serious problem. Despite various authentication methods being used, credit card fraud remains difficult to prevent. Fraud not only can cause significant financial losses for cardholders and banks, but it also affects public trust in the use of credit cards [4].



Therefore, to combat and prevent credit card transaction fraud, effective methods are needed to detect it. One method to identify fraud is by using existing customer data and processing it to identify regularities, patterns, or relationships from a large amount of data using pattern recognition methods such as mathematics and statistics, commonly known as data mining [5]. To perform data mining classification, the steps taken can utilize algorithms found in machine learning. Machine learning focuses on developing a system that can learn and make decisions on its own without needing to be programmed repeatedly by humans [6]. The algorithms commonly used include Random Forest, Naive Bayes, Classification and Regression Trees (CART), Support Vector Machine (SVM), and others. Machine Learning is widely used in fraud detection, marketing targeting, performance prediction, manufacturing, medical diagnostics, and more [7].

In the research titled "Optimization of the Decision Tree Classification Algorithm (CART) with the Bagging Method for Web Phishing Detection," conducted by Ria Ester and Sartika Lina Mulani Sitio in 2024, an accuracy rate of 96.61% was achieved, which increased to 97.74% after optimization with bagging [8]. In the research titled "Prediction System for Cervical Cancer Using CART, Naive Bayes, and k-NN" conducted by Tutus Praningki and Indra Budi in 2017, the accuracy value for Naive Bayes was 94.44%, while CART and k-NN achieved 88.89% and 85.04%, respectively [9]. In the study titled "Email Spam Detection using Naive Bayes and Particle Swarm Optimization" conducted by Nandan Parmar et al. in 2020, the accuracy for Naive Bayes was 87.75% and increased to 95.50% after optimization with PSO [10]. Then, in the research titled "Performance Analysis of CART and Naive Bayes Algorithms Based on Particle Swarm Optimization (PSO) for Cooperative Credit Feasibility Classification" conducted by Eko Arif Riyanto et al. in 2021, the accuracy value for Naive Bayes was 75%, which increased to 96.43% after being combined with PSO, while CART achieved an accuracy of 78.57%, which rose to 92.86% after being combined with PSO [11].

Based on previous research, each algorithm has different accuracy results in its studies. Therefore, the author intends to conduct research with a different object, specifically in detecting fraud in credit card transaction histories, to assess the performance of each algorithm in detecting credit card fraud by referring to studies that have the highest accuracy, whether before or after being optimized with other techniques. Therefore, this research aims to be conducted on the topic of Classification and Regression Trees (CART) and Naive Bayes combined with Particle Swarm Optimization in detecting fraud in credit card transactions.

2. Research Method

2.1. Data Collection

The research data used is a dataset sourced from a public data website that can be searched using the keywords "credit card fraud detection" on the platform Kaggle.com, or the data can be downloaded from the following link: <https://www.kaggle.com/datasets/zeesolver/credit-card>. The data used consists of 568,630 big data credit card transactions, with 31 labels and 2 classes (namely 0 and 1), which indicate fraud or legitimate. The table below represents the research dataset used.

Table 1. Credit Card Transaction Dataset

No	id	V1	V2	...	V27	V28	Amount	Class
0	0	-0.26048	-0.469648	...	-0.081230	-0.151045	17982.10	0
1	1	0.985100	-0.356045	...	-0.248052	-0.064512	6531.37	0
2	2	-0.260272	-0.949385	...	-0.300258	-0.244718	2513.54	0
3	3	-0.152152	-0.508959	...	-0.165316	0.048424	5384.44	0
4	4	0.206820	0.165280	...	0.023712	0.419117	14278.97	0
5	5	0.025302	-0.14051	...	0.066681	0.095812	6901.49	0
6	6	1.016482	-0.397181	...	-0.198163	-0.087619	18954.45	0
...	...	...	...	...	...	...	...	...
568623	568623	-0.87385	0.780667	...	-0.153297	1.860894	11591.09	1
568624	568624	-0.54234	0.585098	...	1.222537	1.172982	10145.64	1
568625	568625	-0.833437	0.061886	...	3.308968	0.081564	4394.16	1
568626	568626	-0.670459	-0.202896	...	-1.528642	1.704306	4653.40	1
568627	568627	-0.311997	-0.004095	...	-0.487540	-0.268741	23572.85	1
568628	568628	0.636871	-0.516970	...	-0.159269	-0.076251	10160.83	1

2.2. Data Preparation

Table 1 presents several parts of the dataset used. In the preparation stage, the process involves cleansing, which means cleaning the data by searching for and correcting (or removing) incomplete, missing, incorrectly formatted, or duplicate data [12], [13]. Data cleansing can be performed by checking for missing or empty values; if any are found, they can be added or removed.

2.3. Split Data

This research was conducted by dividing the data into two parts: training data and testing data, where the training data constitutes 70% of all data, amounting to 398,041 data points, while the testing data makes up 30% of the total data, which is 170,589 data points. The training data is useful for the model to learn from the patterns and features within the training data, while the testing data is used to generate prediction results from each trained model. After that, a class imbalance check was performed using the SMOTE method.

2.4. Algorithm Model

This research utilizes two algorithms, namely Classification and Regression Trees (CART) and Naive Bayes. Additionally, this study also employs a combination with the Particle Swarm Optimization method (PSO).

1. Classification Regression Trees (CART): CART is a nonparametric method for fulfilling classification requirements. This method uses repeated partitioning of the data set into new nodes. The goal of the CART algorithm is to obtain data groups with accurate classification characteristics.
2. Naive Bayes: It is a classification method based on Bayes' theorem that calculates the probability of a target class based on the probabilities of existing features [14]. The goal of this algorithm is to find the best way to compare a portion of new data with a set of classifications in various problems [15].
3. Particle Swarm Optimization (PSO): is one way to improve accuracy by calculating the best weights for each data set and producing an average swarm for the entire sample. The ultimate goal of PSO is to find the optimal solution in the search space. This can be the minimum or maximum value of the objective function being optimized [16], [17].

2.5. Evaluation Model Algorithm

Model evaluation is the process of assessing the performance of a machine learning model to understand how well the model makes predictions based on the given data. The final stage of this research was conducted by involving a confusion matrix, by calculating the accuracy value, precision value, recall value, and AUC value. (Area Under the Curve). The purpose of using a confusion matrix is to show how the classification model performs on the testing dataset with known actual values, allowing us to see how the algorithm operates [18]. The following is the confusion matrix in Table 2.

Table 2. Confusion Matrix

Actual	Prediction	
	Legitimate	Fraudulent
True Legitimate	TP	FN
True Fraudulent	FP	TN

Description:

TP (True Positive): the number of correct predictions that what is predicted as the legitimate class is indeed the positive legitimate class.

TN (True Negative): the number of correct predictions that what is predicted as the fraudulent class is indeed the negative fraudulent class.

FP (False Positive): the number of incorrect predictions that what is predicted as the legitimate class is actually the fraudulent class.

FN (False Negative): the number of incorrect predictions that what is predicted as the fraudulent class is actually the legitimate class.

The performance calculation of the algorithm is done by finding the values of accuracy, precision, recall, and AUC value.

1. Accuracy: Accuracy can indicate the magnitude of the ratio or how accurate the results of a prediction are. The following equation can be used to find the accuracy value.

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} * 100 \quad (1)$$

2. Precision: Precision can be defined as the match between the information request and the response provided. Precision can also be defined as the ratio of relevant items selected compared to all items chosen. To calculate the precision value, you can use the following equation.

$$Negative Precision = \frac{TN}{TN+FN} * 100 \quad (2)$$

$$Positive Precision = \frac{TP}{TP+FP} * 100 \quad (3)$$

$$Overall Precision = \frac{Precision n + Precision p}{2} * 100 \quad (4)$$

3. Recall: Recall is the number of relevant items selected compared to the total number of relevant items available. To calculate the recall value, you can use the following equation.

$$Negative Recall = \frac{TN}{TN+FP} * 100 \quad (5)$$

$$Positive Recall = \frac{TP}{TP+FN} * 100 \quad (6)$$

$$Overall Recall = \frac{recall n + recall p}{2} * 100 \quad (7)$$

4. AUC value: used to measure the performance of predictive models in binary classification. The AUC curve is a square area with values always between 0 and 1, and it indicates the level of accuracy or probability of the predictive model.

3. Method

This research was conducted using an online method with data that can be accessed through an official website that stores various datasets. This research uses a dataset from the website Kaggle.com regarding credit card transaction fraud. The devices used for this research are: 1) hardware including an Acer laptop with Intel Core i5, a minimum of 8GB RAM, and a 1TB HDD. 2) software used includes the Operating System: Microsoft Windows 11, Web Browser: Google Chrome, Microsoft Office: Microsoft Excel 2019, Draw.io, and Python libraries Scikit-Learn, Imblearn, Numpy, Pandas, Matplotlib. The stages of the research conducted include: obtaining the dataset from Kaggle, performing data preprocessing, data splitting, checking for class imbalance using the SMOTE method, training and testing using the CART algorithm, Naive Bayes, PSO + CART, and PSO + Naive Bayes. After obtaining the predictions for fraud and legitimate cases, the performance is then evaluated using a confusion matrix.

4. Result and Discussion

4.1. Test Result

	Actual	Predicted
363369	1	1
158118	0	0
202455	0	0
115521	0	0
193355	0	0
...	...	...
185330	0	0
84822	0	0
447075	1	1
123340	0	0
130355	0	0

[170589 rows x 2 columns]

Fig 1. Comparison of original data and the prediction results of the CART algorithm.

The above figure 1 is the result of predictions from the CART Algorithm compared to the original dataset of the study. The results only display the top 5 data points and the bottom 6 data points because there is a large amount of data available

	Actual	Predicted
363369	1	1
158118	0	0
202455	0	0
115521	0	0
193355	0	0
409024	1	1
206132	0	0
556677	1	1
48935	0	0
538795	1	1
75635	0	0
116853	0	0
493611	1	1
178440	0	0
242041	0	0
537661	1	1
253573	0	0
30533	0	0
42611	0	0
211315	0	0

Fig 2. Comparison of the original data and the prediction results of the Naive Bayes algorithm displayed in 20 data points.

The above figure 2 is the result of predictions from the Naive Bayes Algorithm compared to the original dataset of the study. The results only display the top 20 data points.

```
[58] # Display selected features for Decision Tree
      print("\nSelected Features for Decision Tree using PSO:")
      print(selected_features_dt)
```

Selected Features for Decision Tree using PSO:
[0, 1, 4, 8, 9, 10, 12, 13, 15, 16, 17, 18, 23, 24, 26, 27, 29]

Fig 3. Results of feature selection using PSO for the CART algorithm.

The above figure 3 shows that the selection results conducted by PSO produced 17 relevant attributes for the CART algorithm out of 31 attributes. The selected attributes are id, V1, V4, V8, V9, V10, V12, V13, V15, V16, V17, V18, V23, V24, V26, V27, and the amount.

```
[ ] # Display selected features for Naive Bayes
      print("\nSelected Features for Naive Bayes using PSO:")
      print(selected_features_nb)
```

Selected Features for Naive Bayes using PSO:
[0, 4, 6, 17, 26, 29]

Fig 4. Results of feature selection using PSO for the Naive Bayes algorithm.

The above figure 4 shows that the PSO method only selects 6 attributes deemed relevant for the Naive Bayes algorithm in that dataset. The attributes chosen by PSO for the Naive Bayes algorithm are id, V4, V6, V17, V26, and amount. The prediction results of the algorithm are also displayed in the form of a confusion matrix. Below is the confusion matrix table for each algorithm.

Table 3. Confusion Matrix Algorithm CART

Actual	Prediction	
	Legitimate	Fraudulent
True Legitimate	85264	30
True Fraudulent	33	85262

The data tested consists of 170,589 big data points, which are testing data. The purpose of the table above is as follows: in True Positives (TP), there are 85,264 correct predictions that transactions that are actually legitimate have been accurately classified as legitimate transactions. In False Negatives (FN), there are 30 incorrect predictions, where transactions that are actually legitimate (positive class) have been classified as fraudulent, meaning the model made an error by considering legitimate transactions as fraud. Then, in False Positives (FP), there are 33 incorrect predictions where transactions that are actually fraudulent have been classified as legitimate (negative class). In True Negatives (TN), there are 85,262 correct predictions that transactions that are actually fraudulent have been accurately classified.

Table 4. Confusion Matrix Algorithm Naive Bayes

Actual	Prediction	
	Legitimate	Fraudulent
True Legitimate	75153	10141
True Fraudulent	1545	83750

The meaning of the table above is that, in True Positives (TP), there are 75,153 correct predictions that transactions that are actually legitimate have been correctly classified as legitimate transactions. In False Negatives (FN), there are 10,141 incorrect predictions, where transactions that are actually legitimate (positive class) are classified as fraudulent, meaning the model makes a mistake by considering legitimate transactions as fraud. Then, in False Positives (FP), there are 1,545 incorrect predictions that transactions that are actually fraudulent have been classified as legitimate (negative class). In True Negatives (TN), there are 83,750 correct predictions that transactions that are actually fraudulent have been correctly classified.

Table 5. Confusion Matrix Algorithm CART + PSO

Actual	Prediction	
	Legitimate	Fraudulent
True Legitimate	85264	30
True Fraudulent	29	85266

The meaning of the table above is that, in True Positives (TP), there are 85,264 correct predictions that transactions that are actually legitimate have been correctly classified as legitimate transactions. In False Negatives (FN), there are 30 incorrect predictions, where transactions that are actually legitimate (positive class) are classified as fraudulent; in this case, the model makes a mistake by considering legitimate transactions as fraud. Then, in False Positives (FP), there are 29 incorrect predictions where transactions that are actually fraudulent have been classified as legitimate (negative class). In True Negatives (TN), there are 85,266 correct predictions that transactions that are actually fraudulent have been correctly classified.

Table 6. Confusion Matrix Algorithm Naive Bayes + PSO

Actual	Prediction	
	Legitimate	Fraudulent
True Legitimate	83616	1678
True Fraudulent	888	84407

The meaning of the table above is that, in True Positives (TP), there are 83,616 correct predictions that transactions that are actually legitimate have been correctly classified as legitimate transactions. In False Negatives (FN), there are 1,678 incorrect predictions, where transactions that are actually legitimate (positive class) are classified as fraudulent; in this case, the model makes a mistake by considering legitimate transactions as fraud. Then, in False Positives (FP), there are 888 incorrect predictions that transactions that are actually fraudulent have been classified as legitimate (negative class). In True Negatives (TN), there are 84,407 correct predictions that transactions that are actually fraudulent have been correctly classified.

The following is the performance calculation of the algorithm used in this research.

Table 7. Results of the algorithm performance calculation

Algorithm	Accuracy	Precision	Recall	Auc Value
CART	99,96	100	100	1,00
Naive Bayes	93,15	94	93	0,99
CART + PSO	99,97	100	100	1,00
Naive Bayes + PSO	98,50	98	98	1,00

In addition, the performance results are also presented in the form of graphs and ROC curves.

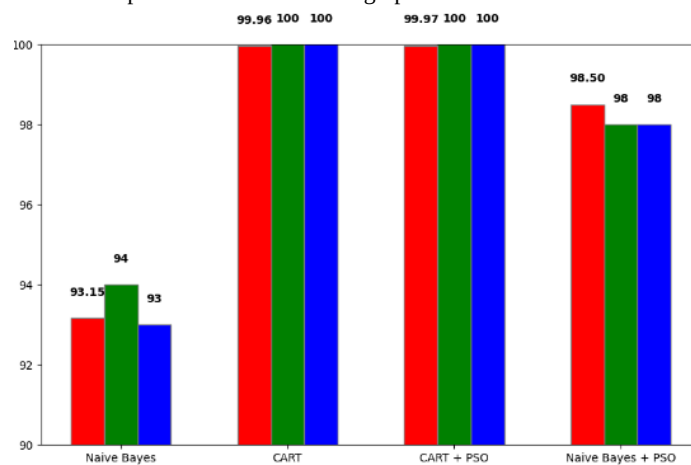


Fig 5. Comparison chart of algorithm performance

The above figure 5 is a comparison graph of the performance of each algorithm. It can be seen from the graph that the CART algorithm combined with PSO has a superior value compared to the other algorithms, although it is only 0.01% different from the CART algorithm without the combination with PSO.

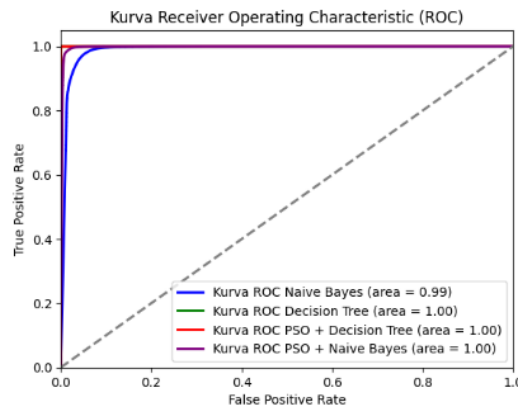


Fig 6. Curve ROC

4.2. Analyze the Test Result

Based on the tests that have been conducted, the performance of the naive bayes algorithm in detecting fraud falls into the good category at 93.15% and becomes increasingly accurate when combined with particle swarm optimization. PSO works very optimally in improving the testing accuracy of the naive bayes algorithm, with data accuracy increasing by 5.35% to 98.50%. This proves that PSO is very suitable to be combined with naive bayes. The CART algorithm also performs very well, achieving a very high result of 99.96%, which is considered nearly perfect. This demonstrates that the performance of the CART algorithm in detecting fraud is very optimal and improves even further when combined with PSO. CART combined with PSO yields an accuracy of 99.97%; although it only increased by 0.01%, the combination with PSO indeed provides a value enhancement. It is also evident in previous research that PSO has a significant impact on improving accuracy, as demonstrated in several studies conducted. PSO identifies the most relevant features for the model, which is useful so that the model is only trained with the most informative features, reducing the risk of overfitting and improving overall accuracy. The dataset used in the research falls into the category of high-dimensional data, meaning it has many features. PSO can handle this high-dimensional data by selecting the most relevant features that can yield better performance. Therefore, the use of PSO combined with the CART and Naive Bayes algorithms shows a significant increase in accuracy compared to using CART and Naive Bayes individually. Based on this, the combination with PSO is the right approach to enhance accuracy in research, and from the four tests that have been conducted, it can be concluded that the CART algorithm combined with PSO is the most optimal method for detecting fraud in big data of credit card transactions.

5. Conclusion

Based on the tests conducted regarding the performance of the Classification and Regression Trees (CART) algorithm and Particle Swarm Optimization-based Naive Bayes for detecting credit card transaction fraud, it can be concluded that the performance testing was carried out through several stages. First is the data analysis stage, which involves understanding the data to identify issues within it. Next, the second stage is data preparation, where data cleaning is performed to check for missing values or to see if there is any data that is missing, followed by the data balancing stage. The third step is to divide the data, allocating 70% for training data and 30% for testing data, where the testing data is used to predict the performance of the algorithm. The next stage involves using a confusion matrix to calculate the accuracy, precision, and recall of the predictions and to determine the best algorithm for detecting credit card transaction fraud. The test results show that the accuracy of the Naive Bayes algorithm is 93.15%, while the Classification and Regression Trees (CART) algorithm has an accuracy of 99.96%. The accuracy of the Naive Bayes algorithm combined with Particle Swarm Optimization (PSO) is 98.50%, and finally, the Classification and Regression Trees (CART) algorithm combined with Particle Swarm Optimization (PSO) has an accuracy of 99.97%. Thus, the best and most optimal model for detecting credit card transaction fraud is the CART algorithm combined with Particle Swarm Optimization (PSO).

References

- [1] K. K. Tangel, D. W. Lumintang, and J. A. Kermite, "Kajian Yuridis Tentang Kartu Bank Dalam Transaksi Perbankan," *Jurnal Fakultas Hukum Universitas Sam Ratulangi*, vol. 12, 2023.
- [2] A. Subroto and A. Arianito, "Penggunaan Kartu Kredit dan Perilaku Belanja Kompulsif: Dampaknya Pada Risiko Gagal Bayar," *Jurnal Manajemen Pemasaran*, vol. 6, no. 1, pp. 1–7, 2011, doi: 10.9744/pemasaran.6.1.1-7.
- [3] N. Christian and J. Veronica, "Dampak Kecurangan Pada Bidang Keuangan Dan Non-Keuangan Terhadap Jenis Fraud di Indonesia," *Jurnal Riset Akuntansi Mercubuana*, vol. 8, no. 1, pp. 91–102, 2022, doi: 10.26486/jramb.v8i1.2401.
- [4] E. Mustika Ginting, E. Saripta Siburian, M. Dwi Syahfitri, J. V Willem Iskandar Pasar, M. Baru Kota Medan, and S. Utara, "Analisis Perilaku Konsumen dan Keamanan Kartu Kredit Perbankan," *Era Mustika Ginting, dkk) Madani: Jurnal Ilmiah Multidisiplin*, vol. 1, no. 4, pp. 2986–6340, 2023, doi: 10.5281/zenodo.7963225.
- [5] A. Dwi Rachman Prabowo and Muljono, "Prediksi Nasabah Yang Berpotensi Membuka Simpanan Deposito Menggunakan Naive Bayes Berbasis Particle Swarm Optimization Prediction of Potential Deposit of Customers Using Naive Bayes-Based Particle Swarm Optimization," *Jurnal Teknologi Informasi*, vol. 17, no. 2, pp. 208–219, 2018, doi: 10.33633/tc.v17i2.1648.
- [6] E. Retnoningsih and R. Pramudita, "Mengenal Machine Learning Dengan Teknik Supervised dan Unsupervised Learning Menggunakan Python," *Bina Insani ICT Journal*, vol. 7, no. 2, pp. 156–165, 2020, doi: 10.51211/biict.v7i2.1422.
- [7] A. Syukron and A. Subekti, "Penerapan Metode Random Over-Under Sampling dan Random Forest untuk Klasifikasi Penilaian Kredit," *Jurnal Informatika*, vol. 5, no. 2, pp. 175–185, 2018, doi: 10.31294/ji.v5i2.4158.

- [8] R. Ester, S. Lina, and M. Sitio, "Optimasi Algoritma Klasifikasi Decision Tree (CART) dengan Metode Bagging untuk Deteksi Web Phising," *Jurnal Jaringan Sistem Informasi Robotik (JSR)*, vol. 8, no. 1, pp. 73–78, 2024, [Online]. Available: <http://ojsamik.amikmitragama.ac.id>
- [9] T. Praningki and I. Budi, "Sistem Prediksi Penyakit Kanker Serviks Menggunakan CART, Naive Bayes, dan k-NN Cervical Cancer Disease Prediction System Using CART, Naive Bayes, and k-NN," *Universitas AM IKOM Yogyakarta*, vol. 4, no. 2, pp. 83–93, 2017.
- [10] N. Parmar and A. Sharma, "Email Spam Detection using Naïve Bayes and Particle Swarm Optimization," *International Journal of Innovative Research in Technology*, vol. 6, no. 10, pp. 367–373, 2020, doi: 10.1109/ICCONS.2018.8662957.
- [11] E. Arif Riyanto, T. Juninisvianty, and D. Ferdian Nasution, "Analisis Kinerja Algoritma CART dan Naive Bayes Berbasis Particle Swarm Optimization (PSO) untuk Klasifikasi Kelayakan Kredit Koperasi," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 8, no. 1, pp. 55–60, 2021, doi: 10.25126/jtiik.202182988.
- [12] F. Putra, H. F. Tahiyat, R. M. Ihsan, R. Rahmaddeni, and L. Efrizoni, "Penerapan Algoritma K-Nearest Neighbor Menggunakan Wrapper Sebagai Preprocessing untuk Penentuan Keterangan Berat Badan Manusia," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 1, pp. 273–281, Jan. 2024, doi: 10.57152/malcom.v4i1.1085.
- [13] D. Barapatre and A. Vijayalakshmi, "Data preparation on large datasets for data science," *Asian Journal of Pharmaceutical and Clinical Research*, vol. 10, pp. 485–488, Apr. 2017, doi: 10.22159/ajpcr.2017.v10s1.20526.
- [14] R. Y. Putri, Z. Yunizar, and S. Safwandi, "Comparison of the Results of the K-Nearest Neighbor (KNN) and Naïve Bayes Methods in the Classification of ISPA Diseases (Case Study: RSUD Fauziah Bireuen)," *JACKA: Journal of Advanced Computer Knowledge and Algorithms*, vol. 1, no. 1, pp. 20–24, 2023, doi: : 10.29103/jacka.v1i1.14535.
- [15] M. Ula and S. Fachrurrazi, "Analisis Sentimen Cyberbullying pada Media Sosial Twitter menggunakan Metode Support Vector Machine dan Naïve Bayes Classifier," *TECHSI - Jurnal Teknik Informatika*, vol. 14, no. 2, pp. 107–123, 2023, doi: 10.29103/techsi.v14i2.12103.
- [16] I. Pakaya and P. Perdana, "Particle Swarm Optimization-Fuzzy Logic Controller Untuk Penyearah Satu Fasa," *Jurnal Ilmiah Edutic*, vol. 1, no. 1, pp. 1–11, 2014, doi: 10.21107/edutic.v1i1.401.
- [17] S. H. Pratiwi, W. Witanti, T. Hendro, U. J. Achmad, and Y. Abstract, "Optimasi Penentuan Vendor Untuk Material Pesawat Menggunakan Algoritma Particle Swarm Optimization," *Jurnal Ilmiah Wahana Pendidikan*, Februari, vol. 2024, no. 4, pp. 825–837, doi: 10.5281/zenodo.10537168.
- [18] I.- Sahputra, B. Bustami, and Cut Farida Aryani, "The Nutritional Classification of Pregnant Women Using Support Vector Machine (SVM)," *Journal Of Informatics And Telecommunication Engineering*, vol. 7, no. 1, pp. 174–182, Jul. 2023, doi: 10.31289/jite.v7i1.9764.