Journal of Intelligent Decision Support System (IDSS) 6 (3) (2023), pp. 138-146

Published by: Institute of Computer Science (IOCS)

Journal of Intelligent Decision Support System (IDSS)

Journal homepage: www.idss.iocspublisher.org

The application of particle swarm optimization (PSO) to improve the accuracy of the naive bayes algorithm in predicting floods in the city of Samarinda

Trisha NurHalisha [1], Faldi[2], Wawan Joko Pranoto[3], Hendra Saputra[4], Asslia Johar Latipah[5], Sayekti Harits Suryawan[6] , Naufal Azmi Verdikha[7]

[1234567] *Informatics Engineering, Muhammadiyah University of East Kalimantan, Samarinda, Indonesia*

Article Info

Article history:

Received Aug 19, 2023
Revised Aug 22, 2023
Accepted Aug 23, 2023

Keywords:

Accuracy
Flood
Naive Bayes,
Optimization
Particle Swarm Optimization (PSO)

ABSTRACT

This study focuses on the implementation of Particle Swarm Optimization (PSO) to enhance the accuracy of the Naive Bayes algorithm in predicting floods specifically in the city of Samarinda. The aim is to improve the efficiency and precision of flood prediction models in order to mitigate the impact of flooding in the area. The results of this research highlight the effectiveness of PSO in optimizing the Naive Bayes algorithm, showing promising potential for more accurate flood prediction and proactive measures in Samarinda. The accuracy value obtained from testing using the Naive Bayes method alone is 91.12%. However, there is an improvement in accuracy after conducting testing with the optimization technique based on Particle Swarm Optimization (PSO) and the Naive Bayes algorithm. The conducted testing achieved an accuracy value of 94.38%. This accuracy result is higher compared to testing without optimization.

Corresponding Author:

Faldi,
Informatics Engineering,
Muhammadiyah University of East Kalimantan,
Jl. Ir. H. Juanda No.15, Sidodadi, Kec. Samarinda Ulu, Kota Samarinda, Kalimantan Timur, 75124, Indonesia.
Email: fal146@umkt.ac.id

Introduction

Badan Nasional Penanggulangan Bencana (BNPB) as The National Disaster Management Agency has reported a total of 3,522 natural disasters occurring in Indonesia throughout the year 2022. Floods were a frequent natural disaster during this year, with a total of 1,520 incidents(Mustajab, 2023). The increasing occurrence of flood disasters can result in various losses, including economic and health-related issues. Several efforts have been made to address the incoming flood disasters, but up until now, the problem of floods has not been effectively resolved.

The provincial capital of East Kalimantan, Samarinda City, is characterized by 27 river streams and low-lying areas with poor water drainage systems. The topographical condition of Samarinda City leads to frequent flood occurrences. High rainfall and rising river water levels are the two main causes of floods in Samarinda City. According to data from the Regional Disaster Management Agency (BPBD) and the Meteorology, Climatology, and Geophysics Agency (BMKG), 75 flood incidents were recorded in Samarinda City between 2019 and 2022. The Naive Bayes algorithm utilized in data mining can provide accurate figures for flood events in Samarinda City, thus necessitating an analysis of flood

disasters(Hasanah et al., 2021). Therefore, it is important to develop effective methods to manage and mitigate the impact of floods.

One of the initial steps in flood management is predicting the probability of flooding in vulnerable areas. The Naive Bayes algorithm has been widely used as a prediction method due to its simplicity and speed. To handle large amounts of data, address missing values, and overcome diverse attribute and data-related issues, the Naive Bayes algorithm is utilized (Yakup, 2020). This algorithm uses Bayes' theorem to calculate the probability of an event occurring based on the occurrence of related events. However, despite the effectiveness of the Naive Bayes algorithm, there are several factors that can affect its accuracy, such as data variability, sample size, and class imbalance. Therefore, a method is needed to improve the accuracy of the Naive Bayes algorithm in predicting floods in Samarinda.

Particle Swarm Optimization (PSO) is a optimization technique that is inspired by social processes within a group. PSO has been widely used in various optimization problems, including classification problems. This technique works by optimizing the positions and velocities of individuals in a population to find the optimal solution.

In the context of flood prediction in Samarinda, the application of Particle Swarm Optimization (PSO) can be used to improve the accuracy of the Naive Bayes algorithm. PSO can assist in determining optimal weights or parameters for Naive Bayes, improving classification learning, and enhancing data representation. By harnessing the power of PSO to search for the best solution, it is expected that the accuracy of flood prediction using the Naive Bayes algorithm can be significantly improved(Naderi et al., 2019; Wiratama & Pradnya, 2022).

Thus, this research aims to apply Particle Swarm Optimization (PSO) in enhancing the accuracy of the Naive Bayes algorithm in flood prediction in Samarinda (Kareem et al., 2020). The results of this research are expected to contribute to the reduction of flood risk and the improvement of flood management in Samarinda. Therefore, to facilitate the analysis, the research utilized a tool for data analysis using the RapidMiner software. RapidMiner is a data science software platform developed by a company of the same name, providing an integrated environment for machine learning, deep learning, text mining, and predictive analytics. This application is used for both business and commercial applications as well as for research, education, training, rapid prototyping, and application development. It supports all steps of the machine learning process, including data preparation, result visualization, validation, and optimization(Nofitri & Irawati, 2019).

Method

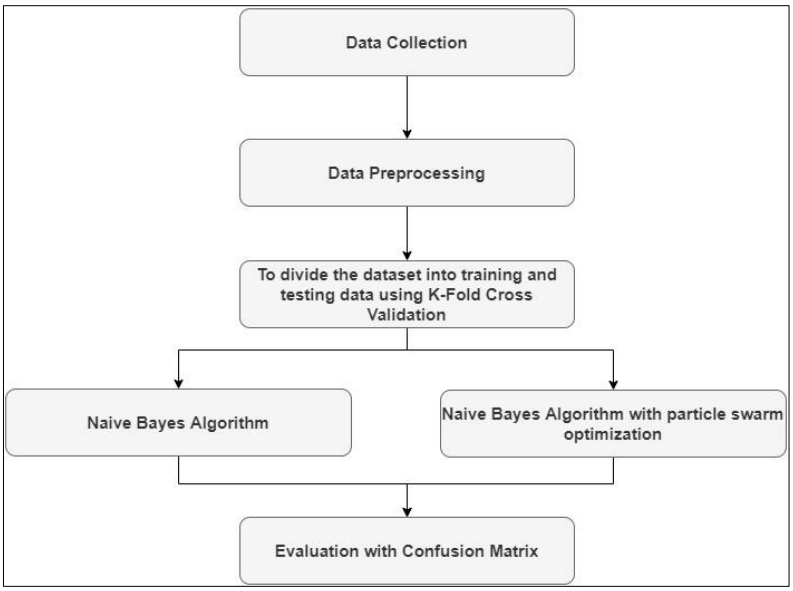

Figure. 1. Research Method

To gather accurate data for problem-solving purposes, during the data collection phase, the researcher obtained flood data by conducting direct observations at the Samarinda City Regional Disaster Management Agency (BPBD) and Meteorology, Climatology, and Geophysics Agency (BMKG) units. Based on the availability of data obtained, spanning from 2019 to the period of 2022, a total of 1461 data points consisting of 20 attributes were collected. The obtained attributes are presented in the following table:

Table 1. Attributes of BMKG and BPBD Data in Samarinda City

Atribut	Description
Tn	Minimum temperature(°C)
Tx	Maximum temperature(°C)
Tavg	Average temperature (°C)
RH_avg	Average Humidity (%)
RR	Rainfall (mm)
ss	Duration of sunshine (hrs)
ff_x	Maximum wind spee (m/s)
ddd_x	Wind direction during maximum speed (°)
ff_avg	Average wind speed (m/s)
ddd_car	Most frequent wind direction (°)
Date	Occurrence time
Flood Occurrence	Yes/ No
Time of Incident	Time of the disaster
Type of Disaster	Type of natural disaster that occurred
Location/Area	Location of the flood
Area Size in Square Meters	Area affected by the flood
Objects Affected by the Disaster	Facilities affected by the disaster
Victims	Number of victims affected by the disaster
Losses	Amount of losses incurred
Description/Remarks	Details of the disaster event

After data collection, the next steps are Business Understanding and Data Understanding(Schröer et al., 2021).

1. Business Understanding: This step involves gaining a thorough understanding of the business problem or objective at hand. It includes identifying the goals, requirements, and constraints of the project. Understanding the business context helps align the data analysis process with the specific needs of the organization or project(Hasanah et al., 2021).
2. Data Understanding: In this step, the collected data is explored to gain insights into its structure, quality, and characteristics. It involves assessing the data's completeness, checking for missing values or outliers, and understanding the relationships between variables. Exploratory data analysis (EDA) techniques, such as summary statistics, data visualization, or correlation analysis, may be used to gain a deeper understanding of the data(Huber et al., 2019).

By performing Business Understanding and Data Understanding, analysts or data scientists can grasp the context of the problem and gain valuable insights into the dataset. This knowledge serves as a foundation for further data preparation, modeling, and analysis in subsequent steps of the data science process.

Data Preprocessing

The Data Preprocessing process involves the following steps(Cazacu & Titan, 2021):

1. Data Selection: After data collection, this step involves selecting the relevant attributes from the BPBD Samarinda and BMKG Samarinda datasets that will be used for analysis.
2. Data Integration: In this step, the data from both BPBD Samarinda and BMKG Samarinda is merged or combined into a single dataset. This integration ensures that all relevant information is available in one consolidated dataset.

3. Data Transformation: This step involves modifying or converting the data to a suitable format or type before using it for modeling. It may include tasks such as normalization, standardization, or encoding categorical variables.
4. Data Cleaning: This step focuses on handling missing values in the dataset. Missing values can be replaced, imputed, or removed based on the analysis and the impact on the overall dataset quality(Venkata & Narsimha, 2021).

By completing these Data Preprocessing steps, the dataset is cleansed, transformed, and ready for further analysis or modeling. These steps help improve data quality, remove inconsistencies, and ensure the dataset is suitable for subsequent stages in the data science workflow.

To divide the dataset

The dataset is divided into two groups: the training data and the testing data. The training data is used to train the model or algorithm, while the testing data is used to evaluate the model by providing input from the testing data and comparing the model's predictions with the actual target values.

In this research, the parameter 'cv = 10' is implemented using the RapidMiner software, and the K-Fold Cross Validation technique is applied. This means that the dataset is divided into 10 subsets or folds, with each fold serving as the testing data once while the remaining folds are used for training. This process is repeated 10 times, rotating which fold is chosen as the testing data. By doing this, it helps to ensure a robust and reliable evaluation of the model's performance by testing it on different subsets of data.

Data Modeling

In this stage, the data classification of flood occurrences in Samarinda City is carried out using the Naive Bayes algorithm, along with the addition of optimization techniques from Particle Swarm Optimization (PSO)(Naderi et al., 2019). The initial step of the Naive Bayes algorithm involves computing the probabilities of classes and attributes. Then, the joint probability of class and attribute is calculated and used in the classification process(Amrin et al., 2021). The Naive Bayes algorithm has the ability to classify new data, and its performance is evaluated by comparing the classification results with the actual class labels. Various evaluation metrics such as accuracy, precision, and recall can be used to measure the performance of this algorithm. Next, the optimization technique of Particle Swarm Optimization (PSO) is applied. The PSO stage starts with initialization, where the population size of particles, initial positions and velocities of each particle, as well as the initialization of the personal best positions (pbest) and global best position (gbest) are determined randomly(Asri et al., 2023). After that, the evaluation stage is performed by calculating the fitness value for each particle based on its current position. Then, the pbest and gbest positions are updated, followed by updating the velocities and positions of particles. The process of convergence evaluation and iteration continues until it produces the output with the best particle position (gbest) that has the optimal fitness value(Ali & Farida, 2021).

Evaluation

In the Evaluation stage, validation and measurement of the accuracy of the obtained results are performed using the CRISP-DM method and the Confusion Matrix to assess the accuracy of the algorithm(Utomo & Mesran, 2020). The CRISP-DM (Cross-Industry Standard Process for Data Mining) method is a widely used framework for conducting data mining projects. It involves various steps, including data understanding, data preparation, model building, evaluation, and deployment. The Evaluation stage in CRISP-DM focuses on measuring the performance and accuracy of the developed model(Ahmad et al., 2021; Dåderman & Rosander, 2018)

One of the commonly used techniques in evaluating the performance of a classification algorithm is the Confusion Matrix(Haghighi et al., 2018). The Confusion Matrix summarizes the performance of the algorithm by providing information about true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. From these values, various metrics such as accuracy, precision, recall, and F1-score can be calculated to assess the algorithm's accuracy (Hasnain et al., 2020; Markoulidakis et al., 2021).

By applying the CRISP-DM methodology and utilizing the Confusion Matrix, a comprehensive

evaluation of the Naive Bayes algorithm with the PSO optimization technique can be conducted to determine its accuracy and effectiveness in classifying flood occurrences in Samarinda City.

Results and Discussions

In the modeling stage using RapidMiner, the Naive Bayes algorithm will be used to measure accuracy. Model testing is conducted using the K-Fold Cross Validation method with a value of K=10. The preparation of the training data begins by importing the data using the "Retrieving Data" operator. The modeling process in RapidMiner can be seen in Figure 2.

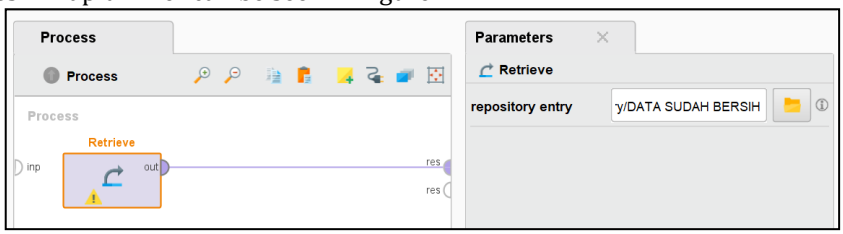

Figure. 2. The modeling stage in RapidMiner

After the preparation of the training data is completed, the next step is to divide the dataset using "Cross Validation" with K=10. This process can be seen in Figure 3

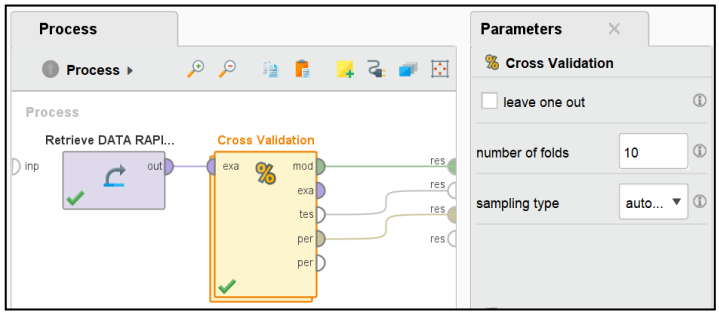

Figure. 3. Cross Validation in RapidMiner

Next, we select the Naive Bayes algorithm as the model for data mining. Testing is performed using the "Apply Model" operator in RapidMiner. Model performance evaluation is done using the "Performance" option. This process can be seen in Figure 4.

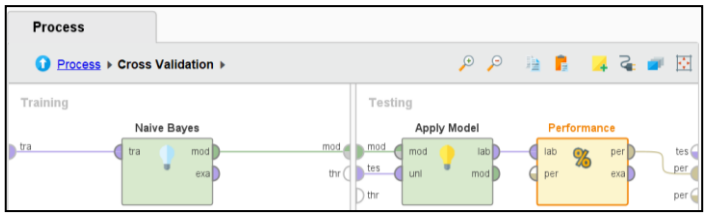

Figure. 4. Choose Algorithm Model and Performance

The result of modeling using Naive Bayes will produce a Confusion Matrix consisting of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). The processed data is inputted into the Confusion Matrix, resulting in the following outcomes as seen in Table 2 below:

Table 2. Confusion Matrix Algoritma Naive Bayes

		Actual	
		1 (True)	0 (False)
Predicted	Positive	914	47
	Negative	43	10

From Table 2, the accuracy result obtained is 91.12%. The manual calculation for accuracy is as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100\% \quad \ldots\ldots (1)$$

$$= \frac{914+10}{914+10+47+43}\ \text{x}\ 100\%$$

$$= \frac{924}{1014}\ x\ 100\%$$

$$= 91{,}1242 = 91{,}12\%$$

With an accuracy of 91.12%, the Confusion Matrix indicates that the algorithm provides correct predictions for 91.12% of the total evaluated data. This can be considered a good indication that this algorithm can be used to predict the likelihood of floods in Samarinda City with a high degree of accuracy.

Naive Bayes Modeling with PSO in RapidMiner

Particle Swarm Optimization (PSO) modeling is used to assign weights to data and enhance the computation results(Sa'diyah et al., 2020). Minimum and maximum values are taken into consideration because these weights are determined randomly. Each particle in the population will have its own weights for the attributes in the dataset, and the Naive Bayes algorithm will be applied to calculate the accuracy level of the formed model(Hayatin et al., 2020). The modeling process with PSO begins by importing data using the "Retrieving Data" operator and selecting Selection - Optimization - Optimize Weight (PSO) to optimize the attribute weights by applying the Particle Swarm Optimization (PSO) algorithm(Putri et al., 2020). The visualization of this process can be seen in Figure 5.

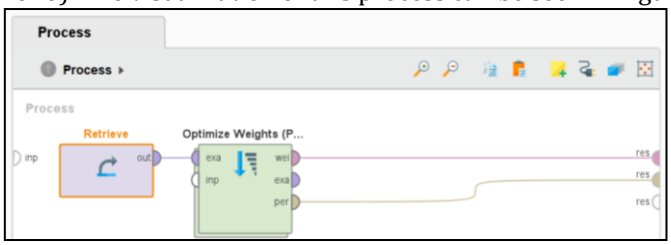

Figure. 5. Selection of Optimization

The next step is to divide the dataset using the "Cross Validation" technique with a value of K=10. The RapidMiner interface for this process can be seen in Figure 6.

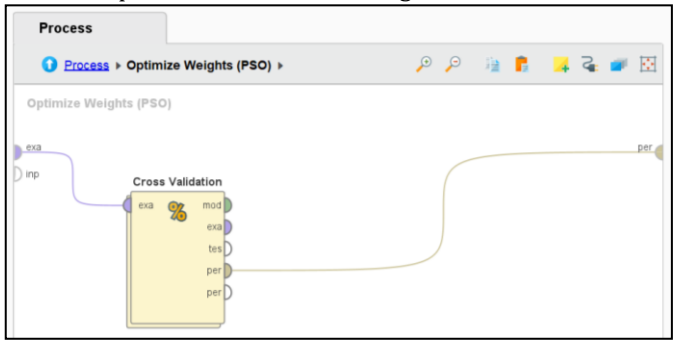

Figure. 6. Cross Validation

Afterwards, the next step is to select the algorithm model for data mining. In this research, the chosen model is Naive Bayes. To test this model, the "Apply Model" operator is used in RapidMiner. Furthermore, to evaluate its performance, the "Performance" option is utilized. The RapidMiner interface for this process can be seen in Figure 7.

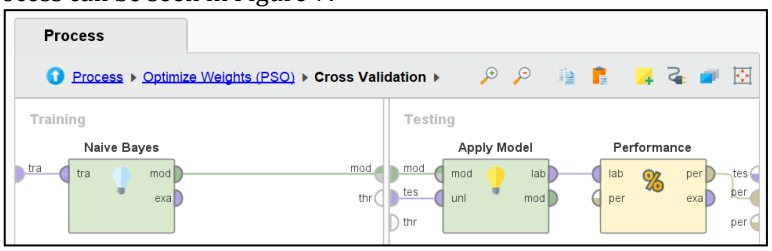

Figure. 7. Model Algorithm Selection and Performance Evaluation Process

The outcome of the modeling process will yield a Confusion Matrix consisting of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values. The processed data is inserted into the Confusion Matrix to obtain the results shown in Table 3 below:

Table 3. Confusion Matrix for Naive Bayes Algorithm with PSO in RapidMiner

		Actual	
		1 (True)	0 (False)
Predicted	Positive	957	57
	Negative	0	0

From Table 3, the Confusion Matrix yields an accuracy of 94.38%. The manual calculation for accuracy is as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100\% \quad \text{..... (2)}$$

$$= \frac{957+0}{957+0+57+0}\ \text{x}\ 100\%$$

$$= \frac{957}{1014}\ x\ 100\%$$

$$= 94{,}3786 = 94{,}38\%$$

In this study, after applying Particle Swarm Optimization (PSO), the calculation result of the Confusion Matrix shows that the implemented algorithm achieves an accuracy rate of 94.38%. This indicates a significant improvement in the algorithm's ability to predict floods in Samarinda City.

Conclusions

In this study, Particle Swarm Optimization (PSO) was implemented to enhance the accuracy of the Naive Bayes algorithm in flood prediction in Samarinda City. The analysis results indicate that the use of PSO successfully improved the Naive Bayes algorithm's accuracy from 91.12% to 94.38%. The application of PSO assists in finding optimal weights or parameters for the Naive Bayes algorithm. By optimizing the positions and velocities of individuals in the population, PSO aids in finding better parameter settings, which have proven to enhance the algorithm's capability in predicting flood occurrences. The Confusion Matrix calculations demonstrate that the algorithm provides accurate predictions with a high level of accuracy. The 94.38% accuracy rate can be considered excellent in predicting floods in Samarinda City. The results of using the algorithm revealed that the data attributes that significantly affect the occurrence of floods are the wind direction during maximum speed and average wind speed. This brings

significant benefits to flood management, enabling authorities to take more effective preventive and mitigation measures. This research makes an important contribution to improving flood prediction in Samarinda City and reducing the associated risks of flood disasters. However, continuous evaluation, monitoring, and further improvements are necessary to achieve even higher levels of accuracy. For future research, it is recommended to conduct experiments and validations using a larger dataset covering a longer period. Additionally, other optimization techniques can be explored, and other factors that may influence flood prediction, such as weather, topography, and real-time data availability, should be considered. By continuously advancing the methodology and techniques in flood prediction, it is hoped that mitigation efforts and flood management in Samarinda City can be sharpened, ultimately safeguarding the community and promoting the city's sustainable growth.

References

Ahmad, A., Sakidin, H., Sari, M. Y. A., Amin, A. R. M., Sufahani, S. F., & Rasib, A. W. (2021). Naïve Bayes Classification of High-Resolution Aerial Imagery. *International Journal of Advanced Computer Science and Applications*, *12*(11), 168–177. https://doi.org/10.14569/IJACSA.2021.0121120

Ali, M., & Farida, B. N. I. (2021). Completion of FCVRP using Hybrid Particle Swarm Optimization Algorithm. *Jurnal Teknik Industri*, *22*(1), 98–112. https://doi.org/10.22219/jtiumm.vol22.no1.98-112

Amrin, A., Pahlevi, O., & Satriadi, I. (2021). Optimasi Algoritma C4. 5 dan Naïve Bayes Berbasis Particle Swarm Optimization Untuk Diagnosa Penyakit Peradangan Hati. *INSANTEK-Jurnal Inovasi Dan Sains Teknik Elektro*, *2*(1), 10–14.

Asri, A. M., Ahmad, S. R., & Yusop, N. M. M. (2023). Feature Selection using Particle Swarm Optimization for Sentiment Analysis of Drug Reviews. *International Journal of Advanced Computer Science and Applications*, *14*(5), 286–295. https://doi.org/10.14569/IJACSA.2023.0140530

Cazacu, M., & Titan, E. (2021). Adapting CRISP-DM for social sciences. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, *11*(2Sup1), 99–106.

Dåderman, A., & Rosander, S. (2018). *Evaluating frameworks for implementing machine learning in signal processing: A comparative study of CRISP-DM, SEMMA and KDD*.

Haghighi, S., Jasemi, M., Hessabi, S., & Zolanvari, A. (2018). PyCM: Multiclass confusion matrix library in Python. *Journal of Open Source Software*, *3*(25), 729.

Hasanah, M. A., Soim, S., & Handayani, A. S. (2021). Implementasi CRISP-DM Model Menggunakan Metode Decision Tree dengan Algoritma CART untuk Prediksi Curah Hujan Berpotensi Banjir. *Journal of Applied Informatics and Computing*, *5*(2), 103–108. https://doi.org/10.30871/jaic.v5i2.3200

Hasnain, M., Pasha, M. F., Ghani, I., Imran, M., Alzahrani, M. Y., & Budiarto, R. (2020). Evaluating trust prediction and confusion matrix measures for web services ranking. *IEEE Access*, *8*, 90847–90861.

Hayatin, N., Marthasari, G. I., & Nuarini, L. (2020). Optimization of Sentiment Analysis for Indonesian Presidential Election using Naïve Bayes and Particle Swarm Optimization. *Jurnal Online Informatika*, *5*(1).

Huber, S., Wiemer, H., Schneider, D., & Ihlenfeldt, S. (2019). DMME: Data mining methodology for engineering applications–a holistic extension to the CRISP-DM model. *Procedia Cirp*, *79*, 403–408.

Kareem, T. A., Hussain, M. A., & Jabbar, M. K. (2020). Particle swarm optimization based beamforming in massive MIMO systems. *International Journal of Interactive Mobile Technologies*, *14*(5), 176–192. https://doi.org/10.3991/IJIM.V14I05.13701

Markoulidakis, I., Kopsiaftis, G., Rallis, I., & Georgoulas, I. (2021). Multi-class confusion matrix reduction method and its application on net promoter score classification problem. *The 14th Pervasive Technologies Related to Assistive Environments Conference*, 412–419.

Mustajab, R. (2023). *BNPB: Indonesia Alami 3.522 Bencana Alam pada 2022*. Dataindonesia.Id.

Naderi, E., Pourakbari-Kasmaei, M., & Abdi, H. (2019). An efficient particle swarm optimization algorithm to solve optimal power flow problem integrated with FACTS devices. *Applied Soft Computing*, *80*, 243–262.

Nofitri, R., & Irawati, N. (2019). Analisis Data Hasil Keuntungan Menggunakan Software Rapidminer. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, *5*(2), 199–204.

Putri, D. A., Kristiyanti, D. A., Indrayuni, E., Nurhadi, A., & Hadinata, D. R. (2020). Comparison of naive bayes algorithm and support vector machine using pso feature selection for sentiment analysis on e-wallet review. *Journal of Physics: Conference Series*, *1641*(1), 12085.

Sa'diyah, N., Supianto, A. A., & Dewi, C. (2020). Implementasi Algoritme Fuzzy C-Means dengan Particle Swarm Optimization (FCMPSO) untuk Pengelompokan Proses Berpikir Siswa dalam Proses Belajar. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *4*(6), 1625–1632.

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A systematic literature review on applying CRISP-DM process model.

Procedia Computer Science, *181*, 526–534.

Utomo, D. P., & Mesran, M. (2020). Analisis komparasi metode klasifikasi data mining dan reduksi atribut pada data set penyakit jantung. *Jurnal Media Informatika Budidarma*, *4*(2), 437–444.

Venkata, R. B., & Narsimha, G. (2021). A Multi-purpose Data Pre-processing Framework using Machine Learning for Enterprise Data Models. *International Journal of Advanced Computer Science and Applications*, *12*(3), 646–656. https://doi.org/10.14569/IJACSA.2021.0120376

Wiratama, M. A., & Pradnya, W. M. (2022). Optimasi algoritma data mining menggunakan backward elimination untuk klasifikasi penyakit diabetes. *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, *11*(1), 1–12.

Yakup, S. (2020). Diabetes Mellitus Detection Expert System Using a WEB-Based Naïve Bayesian Approach. *Journal of Intelligent Decision Support System (IDSS)*, *3*(2), 51–61.