Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Feature Selection in Naïve Bayes for Predicting ICU Needs of COVID-19 Patients

Taslim[1], Fajrizal[2], Susi Handayani[3], Dafwen Toresa[4]
taslim@unilak.ac.id, fajrizal@unilak.ac.id, susi@unilak.ac.id, dafwen@unilak.ac.id
[1,2,3,4]Faculty of Computer Science, Universitas Lancang Kuning, Pekanbaru, Riau, Indonesia

Article Information

Submitted : 20 May 2023
Reviewed : 21 Jun 2023
Accepted : 27 Jun 2023

Keywords

Covid-19, prediction Naïve Bayes, Prediction, PSO. accuracy

Abstract

COVID-19 is a global pandemic that requires a coordinated global response in all healthcare and national healthcare systems. Identifying patients at high risk of contracting the COVID-19 virus is crucial to increasing awareness before patients become further infected by the virus, which can cause severe respiratory illnesses requiring specialized care in intensive care units (ICUs). This study aims to predict the need for ICUs in patients infected with the COVID-19 virus. The predicted ICU requirements serve as a reference for hospitals to meet the ICU needs of COVID-19 patients. The prediction of ICU requirements for COVID-19 patients is performed using the Naïve Bayes algorithm, and particle swarm optimization (PSO) used to obtain the best accuracy values from Naïve Bayes. In the initial testing, Naïve Bayes without feature selection resulted in an accuracy rate of 74.75%. Testing Naïve Bayes+PSO by increasing the number of PSO generations shows that as the number of generations in PSO increases, the accuracy rate also increases. Testing Naïve Bayes+PSO with 3000 generations and a population size of 20 shows an increase in the accuracy rate to 80.95%. Testing Naïve Bayes+PSO by increasing the population size to 40 with 1000 generations for each population size shows an increase in the accuracy rate to 80.70%.

A. Introduction

COVID-19 can cause severe respiratory illnesses that may require treatment in intensive care units (ICUs) [1]. Therefore, an early prediction of patients who are likely to require ICU care is needed to improve treatment management and reduce the risk of death. One form of machine learning application is to make predictions based on available data by learning patterns from a dataset and applying them to unknown data to predict outcomes. Classification is one of the techniques in machine learning widely used for data prediction[2].

In machine learning and artificial intelligence, there are several algorithms for making predictions, and one of them is the naive Bayes algorithm. Naive Bayes has proven to be a simple and efficient method for classification in multivariate analysis[3]. This algorithm can be used to make predictions influenced by or affecting probability fluctuations. The algorithm can effectively handle both continuous and discrete data and is not affected by unrelated features [4]. Naive Bayes classification is based on the assumption that attribute values are independent of each other, but this does not always yield satisfactory results. Therefore, many studies have been conducted to improve the performance of the naive Bayes algorithm[5]. One of the methods is by perform feature selection on the data. By reducing excessive and irrelevant data, feature selection offers an efficient solution that can speed up computation, improve learning accuracy, and provide a deeper understanding of the model or learning data [6].

Recently, evolutionary computation methods such as genetic algorithms (GA) and particle swarm optimization (PSO) have been widely used in selecting the best features[7]. This paper analyzes the accuracy of PSO in the Naive Bayes algorithm by comparing their performance in feature selection for COVID-19 patient data requiring ICU.

1. Naive Bayes

Naive Bayes (NB) is one of the data mining algorithms that falls into the category of the 10 popular classification algorithms [8]. This algorithm is known for its simplicity and the assumption that each attribute is independent of the others, where each attribute is treated equally [9]. The Bayes' theorem can be formulated as follows:

$$P(Ci|X) = \frac{P(X|Ci)P(Ci)}{P(X)}$$

Explanation:

- X : Data from an unknown class
- Ci : Hypothesis that the data belongs to a specific class
- P(Ci|X) : Posterior probability based on the value of X
- P(Ci) : Prior probability of hypothesis H
- P(X|Ci) : Probability of X given the condition of Hypothesis H
- P(X) : Probability of X

Naive Bayes follows the following workflow:

1. Create a data tuple and its associated class, X = (x1, x2,..., xn).

2. For each class C1, C2,..., Cm, input the tuple X and then perform classification to predict the value of X based on the highest posterior probability. Naive Bayes predicts whether the tuple X belongs to class Ci if

$$P(C_i \mid X) > P(C_j \mid X) \quad \text{for} \quad 1 \leq jm, j \neq i$$

Maximize the value of P(Ci | X). The class Ci for which P(Ci | X) is maximized is called the maximum posteriori hypothesis.

$$P(Ci|X) = \frac{P(X|Ci)P(Ci)}{P(X)}$$

3. The value of P(X) remains constant for all classes, only the value of P(X|Ci)P(Ci) needs to be maximized.
4. To reduce the computational process of evaluating P(X|Ci), Naive Bayes makes the assumption of class independence, which means that the values of the attributes are independent of each other.

$$P(X \mid Ci) = \prod_{k=1}^{n} P(x_k|Ci)$$

$$= P(x_1|C_i)\ x\ P(x_1|C_i)\ x\ \ldots\ \ldots.\ x\ P(x_1|C_i)$$

2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a popular algorithm known for its strong optimization capabilities and simplicity of implementation[10]. It is an effective optimization algorithm inspired by the behavior of bird flocks, where population topology plays a key role in the PSO algorithm [11]. PSO is inspired by the beautiful flight formations of bird flocks, where they fly together in a synchronized manner and can suddenly change direction and regroup into optimal formations.

PSO is an optimization technique where potential solutions are continuously evaluated using a specific quality criterion. The algorithm optimizes problems by moving particles or expected solutions within the problem space using specific functions of the particles' positions and velocities. The movement of a particle is influenced by its own best solution and, generally, the best solutions available from other particles. This collection of particles is called a swarm, and the swarm continuously moves towards the best solution.

There are two topologies used to describe the interactions between particles in PSO: the ring topology and the star topology. In the ring topology, each particle is connected to two other particles with a fixed neighborhood size of 3. On the other hand, in the star topology, a particle is connected to all other particles. Here is an illustration of the topologies in PSO (Figure 1).

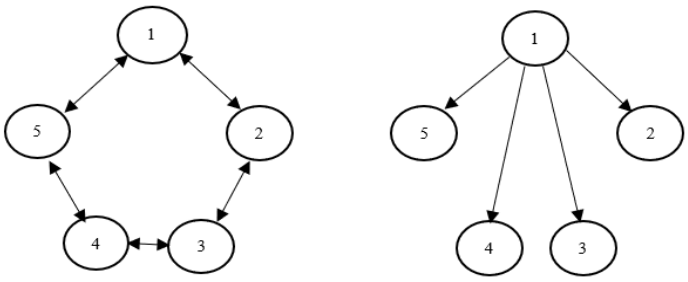

Figure 1: Ring and Star Topologies in PSO

The search process of the PSO algorithm is carried out by a swarm of particles that are updated from iteration to iteration to find the optimal solution. Each particle moves towards its own best position (pbest) encountered so far and the global best position (gbest) discovered by the entire swarm.

$$pbest(i,t) = \arg\ min\left[f\left(Pi(k)\right)\right], \qquad i \in \{1,2,\ldots..Np\},$$

$$gbest(t) = \arg\ min\left[f\left(Pi(k)\right)\right],$$

Explanation:
- I : Represents the index value of a particle.
- Np : Indicates the total number of particles in the swarm.
- t : Refers to the current iteration number.
- f : Represents the fitness function used to evaluate the quality of a solution.
- P : refers to the position of a particle in the search space.

The velocity V and position P of a particle are updated using the following equations:

$$Vi(t+i) = \omega Vi(t) + c_1 r_1 \left(Pbest(i,t)\right) + c_2 r_2 \left(gbest(i,t)\right) - (Pi,t))$$
$$Pi(t+1) = Pi(t) + Vi(t+1)$$

Explanation:
- V: Velocity of the particle
- ω (omega): inertia weight
- r1 and r2: uniformly distributed random variables
- c1 and c2: positive constant parameters

B. Research Method

The classification model in this research uses a COVID-19 patient dataset that contains class labels indicating the need for ICU. Class label 0 represents the need for ICU, while class label 1 represents no need for ICU. The total number of data points used in this research is 10,000, with 5,000 data points for class label 0 and 5,000 data points for class label 1. In this study, testing is conducted through several experiments to find the optimal features from a set of COVID-19 patient data features.

1. Naïve Bayes Experiment without Feature Selection

In this experiment, the data will be split into 80% training data and 10% testing data.

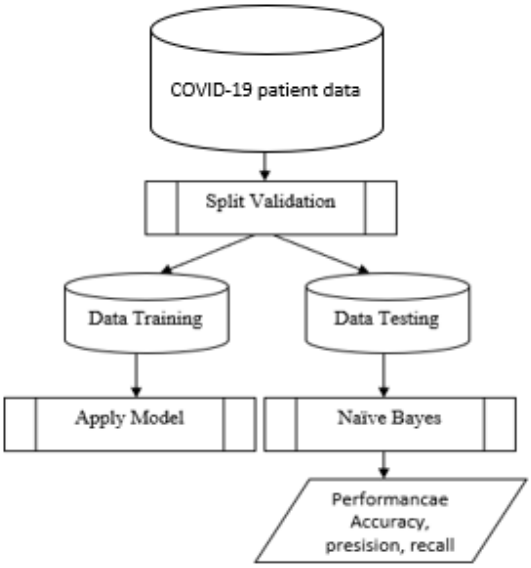

Figure 2. Naïve Bayes algorithm without feature selection

2. Experiment of Feature Selection with Naïve Bayes Using PSO
   2.1. Experiment to evaluate the influence of PSO iterations on accuracy.
   2.2. Experiment to evaluate the influence of PSO population size (number of particles) on accuracy.

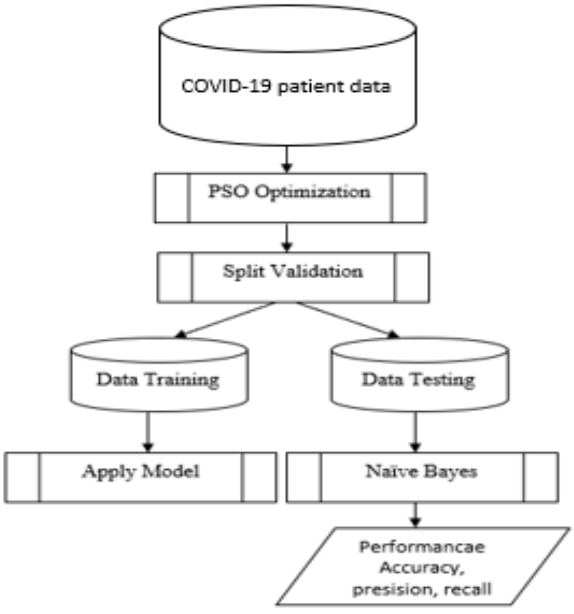

Figure 3. Naïve Bayes algorithm with PSO feature selection

C. Result and Discussion

1. Naïve Bayes Without Feature Selection

The prediction of patient recovery time using Naïve Bayes involves all 18 features of the COVID-19 patient data, with two class labels: 0 and 1. From the testing results, an accuracy of 74.75% was obtained. Table 1 below shows the confusion matrix results of Naïve Bayes without feature selection.

Table 1. Confusion Matrix of Naïve Bayes

		Predicted		Class precision
		Class 0	**Class 1**	
Actual	Class 0	562	67	89,35%
	Class 1	5	72	68,05%
Class Recall		56.20%	93.30%	

2. The experiment conducted in this study involves feature selection with Naïve Bayes using Particle Swarm Optimization (PSO).

2.1. Experiment on the effect of the number of generations in PSO on accuracy.

In this study, an initial population size of 20 populations was used. This is based on the research by Kennedy and Eberhart in their early publication on PSO, which simulated 15–30 bird flocks in zoology research. The authors used 20 particles in their initial PSO testing, and this served as the basis for the initial choice of the number of generations in PSO[20]. The maximum number of generations tested was 1000, 2000, and 3000. The values of inertia weight, local best weight, and global best weight were set to 1.0. The results of the experiments can be seen in Table 2. Flowchart of Naïve Bayes+PSO with feature selection.

Table 2. Results of testing the number of PSO generations on accuracy level

Population Size	Number of generations	Accuracy
20	1000	80.60%
20	2000	80.60%
20	3000	80.95%

The confusion matrix results for each test can be seen in Tables 3, 4, and 5 below.

Table 3. Confusion Matrix of Naïve Bayes + PSO with 20 populations and 1000 generations

		Predicted		Class precision
		Class 0	**Class 1**	
Actual	Class 0	685	73	90.37%
	Class 1	315	927	74,64%
Class Recall		68.50%	92.70%	

Table 4. Confusion Matrix of Naïve Bayes + PSO with 20 populations and 2000 generations

		Predicted		Class precision
		Class 0	**Class 1**	
Actual	Class 0	685	73	90.37%
	Class 1	315	927	74,64%
Class Recall		68.50%	92.70%	

Table 5. Confusion Matrix of Naïve Bayes + PSO with 20 populations and 3000 generations

		Predicted		Class precision
		Class 0	**Class 1**	
Actual	Class 0	688	69	90.89%
	Class 1	312	931	74,90%
Class Recall		68.80%	93.10%	

Table 6. Weight of Each Attribute or Feature

attribut	**Weight**		
	1000 generation	**2000 generation**	**3000 generation**
sex	1.0	0.3	0.0
patient_type	1.0	1.0	1.0
intubed	1.0	1.0	1.0
pneumonia	1.0	1.0	1.0
age	1.0	1.0	1.0
pregnancy	1.0	1.0	1.0
diabetes	0.0	0.0	0.0
copd	0.0	0.0	0.0
asthma	0.0	0.0	0.0
inmsupr	0.0	0.0	0.0
hypertension	1.0	1.0	1.0
other_disease	0.0	0.0	0.0
cardiovascular	0.0	0.0	0.0
obesity	1.0	1.0	1.0
renal_chronic	0.0	0.0	0.0
tobacco	0.0	0.0	0.0
contact_other_covid	1.0	1.0	1.0
covid_res	1.0	0.9	1.0

From Table 6, it can be observed that as the number of generations increases, the accuracy level also increases. From Table 8, it is also evident that the number of generations has an impact on the feature selection results. As the number of

generations increases, the selected features decrease. The influence of the number of generations on the feature selection results can be seen in Table 7 below.

Table 7. Influence of population size and number of generations on feature selection

Population size	Number of generations	Number of selected feature
20	1000	10
20	2000	10
20	3000	9

2.2. Experiencing the Influence of Population Size (Number of Particles) on Accuracy

In this experiment, the influence of population size and the number of generations on feature selection will be tested. The selected population sizes are 20, 30, and 40, with a fixed number of generations set to 1000. The results of the experiment can be seen in Table 8 below.

Table 8. Results of the experiment on the influence of PSO popsize on accuracy level

Population size	Number of generations	Accuracy
20	1000	80.60%
30	1000	80.70%
40	1000	81.10%

The confusion matrix results from each experiment with a population size of 20, 30, and 40, and 1000 generations can be seen in Table 9, Table 10, and Table 11, respectively.

Table 9. Confusion Matrix of Naïve Bayes + PSO with a population size of 20 and 1000 generations.

		Predicted		Class precision
		Class 0	Class 1	
Actual	Class 0	685	73	90.37%
	Class 1	315	927	74,64%
Class Recall		68.50%	92.70%	

Table 10. Confusion Matrix of Naïve Bayes + PSO with a population size of 30 and 1000 generations.

	Predicted		Class precision
	Class 0	Class 1	

Actual	Class 0	673	59	91.94%
	Class 1	327	941	72,21%
Class Recall		67.50%	94.10%	

Table 11. Confusion Matrix of Naïve Bayes + PSO with a population size of 40 and 1000 generations

		Predicted		**Class precision**
		Class 0	**Class 1**	
Actual	Class 0	704	82	89.57%
	Class 1	296	918	75.62%
Class Recall		70.40%	92.80%	

For the weights of each attribute or feature, please refer to Table 12 below.

Table 12. Attribute weights with a maximum of 1000 generations

attribut	**Population size**		
	20	**30**	**40**
sex	1.0	0.0	0.0
patient_type	1.0	0.02	1.0
intubed	1.0	1.0	1.0
pneumonia	1.0	1.0	0.0
age	1.0	0.0	0.0
pregnancy	1.0	0.0	0.0
diabetes	0.0	0.0	0.0
copd	0.0	1.0	1.0
asthma	0.0	0.0	0.0
inmsupr	0.0	0.0	1.0
hypertension	1.0	1.0	0.0
other_disease	0.0	0.0	0.0
cardiovascular	0.0	1.0	0.0
obesity	1.0	1.0	0.0
renal_chronic	0.0	0.0	0.0
tobacco	0.0	0.0	0.0
contact_other_covid	1.0	1.0	1.0
covid_res	1.0	0.2	0.0

From Table 12, it can be observed that as the number of generations increases, the accuracy level also tends to increase. Similarly, from Table 8, it can be seen that the number of generations also influences the feature selection results. With a higher number of generations, the selected features become fewer. The impact of the number of generations on feature selection results can be seen in Table 13.

Table 13. The Influence of PSO Population Size and Number of Generations on Feature Selection

Population size	Number of generations	Number of selected feature
20	1000	10
30	1000	8
40	1000	5

3. Conclusion

From the test results, it can be observed that Swarm Intelligence for feature selection in Naïve Bayes can improve the accuracy rate compared to Naïve Bayes without feature selection. The initial test of Naïve Bayes without feature selection yielded an accuracy rate of 74.75%.

The first test of Naïve Bayes+PSO by increasing the number of PSO generations showed that as the number of generations in PSO increased, the accuracy rate also increased. The first test of Naïve Bayes+PSO with 1000, 2000, and 3000 generations and a population size of 20 showed an improvement in the accuracy rate to 80.60%, 80.60%, and 80.95% respectively. Similarly, the results of the Naïve Bayes+PSO test by increasing the population size to 20, 30, and 40 with 1000 generations for each population size showed an increase in the accuracy rate.

4. References

[1] P. Radanliev, D. De Roure, dan R. Walton, "Data mining and analysis of scientific research data records on Covid 19 mortality , immunity , and vaccine development - in the first wave of the Covid-19 pandemic," 2020.

[2] C. B. C. Latha dan S. C. Jeeva, "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques," *Informatics Med. Unlocked*, vol. 16, no. June, hal. 100203, 2019, doi: 10.1016/j.imu.2019.100203.

[3] R. Blanquero, E. Carrizosa, dan P. Ramírez-cobo, "Computers and Operations Research Variable selection for Naïve Bayes classification," *Comput. Oper. Res.*, vol. 135, hal. 105456, 2021, doi: 10.1016/j.cor.2021.105456.

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

[5] G. Kopanitsa, "ScienceDirect ScienceDirect Performance Improvement Algorithms Algorithms in in Big Big Data Data Analysis Analysis Performance Improvement," *Procedia Comput. Sci.*, vol. 178, hal. 386–393, 2020, doi: 10.1016/j.procs.2020.11.040.

[6] J. Cai, J. Luo, S. Wang, dan S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, hal. 70–79, 2018, doi: https://doi.org/10.1016/j.neucom.2017.11.077.

[7] P. Hu, J.-S. Pan, S.-C. Chu, dan C. Sun, "Multi-surrogate assisted binary particle swarm optimization algorithm and its application for feature selection," *Appl. Soft Comput.*, vol. 121, hal. 108736, 2022, doi: https://doi.org/10.1016/j.asoc.2022.108736.

[8] H. Zhang, L. Jiang, dan L. Yu, "Attribute and instance weighted naive Bayes," *Pattern Recognit.*, vol. 111, hal. 107674, 2021, doi: https://doi.org/10.1016/j.patcog.2020.107674.

[9] S. Chen, G. I. Webb, L. Liu, dan X. Ma, "A novel selective naïve Bayes algorithm," *Knowledge-Based Syst.*, vol. 192, hal. 105361, 2020, doi: https://doi.org/10.1016/j.knosys.2019.105361.

[10] R. Wang, K. Hao, L. Chen, T. Wang, dan C. Jiang, "A novel hybrid particle swarm optimization using adaptive strategy," *Inf. Sci. (Ny).*, vol. 579, hal. 231–250, 2021, doi: https://doi.org/10.1016/j.ins.2021.07.093.

[11] J. Peng, Y. Li, H. Kang, Y. Shen, X. Sun, dan Q. Chen, "Impact of population topology on particle swarm optimization and its variants: An information propagation perspective," *Swarm Evol. Comput.*, vol. 69, hal. 100990, 2022, doi: https://doi.org/10.1016/j.swevo.2021.100990.