Implementation of PPCA Imputation, SMOTE-N Class Balancing in Hepatitis Classification Using Naïve Bayes

Siti Fathmah[1], Dwi Kartini[2]*, Friska Abadi[3], Irwan Budiman[4], M Itqan Mazdadi[5]

[1,2,3,4,5]*Computer Science, Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University, Indonesia*

*corr_author: dwikartini@ulm.ac.id

Abstract - The availability of complete data in research is crucial, especially in the initial stages. The Hepatitis data used in this study encountered issues such as missing data and class imbalance, which hindered its optimal utilization. The method employed to address missing data was the PPCA imputation method. After filling in the missing data, the data was balanced using the SMOTE-N class balancing method and classified using Gaussian Naïve Bayes. The aim of this research was to compare the classification evaluation of hepatitis disease using Naive Bayes with the PPCA imputation approach and SMOTE-N class balancing. The best results from each scenario yielded an AUC value of 0.833 in the first scenario with an 80:20 data split for training and testing, and 0.875 in the second scenario with a 90:10 data split. The highest AUC value was obtained in the application of PPCA imputation with SMOTE-N class balancing using Naive Bayes classification. This demonstrates that the implementation of PPCA imputation with SMOTE-N class balancing has a better impact on the performance of Naïve Bayes classification.

Keywords: classification, naïve bayes, ppca, smote-n, hepatitis

I. INTRODUCTION

Hepatitis is a global health issue that causes fatalities across various age groups, from infants to the elderly. According to WHO data, hepatitis viruses accounted for 1.34 million deaths worldwide in 2015 [1], [2]. With the advancement of information technology, the utilization of hepatitis data is often conducted, including leveraging data mining algorithms. Data mining algorithms are a series of steps useful in processing information from large and complex datasets efficiently, especially in classification tasks [3].

Classification is a data mining functionality that generates models to predict classes or categories of objects within a database [4]. Gaussian Naive Bayes (GNB) is a statistical classification method used to predict the probability of membership in a class [5]. GNB is straightforward, easy to implement, and does not require extensive training data [6]. The classification ability of Naïve Bayes in classifying hepatitis data has been previously studied by Husniah et al [7] an accuracy of 88.52% was obtained from the confusion matrix. Another study conducted by Febrian et al [8] compared the classification results of Naïve Bayes with KNN using the Pima Indian Diabetes dataset obtained from Kaggle. The evaluation results showed that Naïve Bayes outperformed KNN classification, with an accuracy of 76.07% compared to KNN's 73.33%. Another study was conducted by Derisma [9] compared the classification methods of Random Forest, Naïve Bayes, and Neural Network. The research demonstrated that the Naïve Bayes classification model outperformed the other methods with an AUC value of 0.90.

One drawback of the Gaussian Naive Bayes classification method is its inability to effectively classify data with missing values [10]-[12]. Missing values can reduce data accuracy and quality [13]. One approach to handle missing values is imputation, with Probabilistic Principal Component Analysis (PPCA). Study by Hegde et al [14] identified as a more efficient method compared to MICE, achieving a percentage of 65% versus 38%, respectively. Another issue that can affect the Gaussian Naive Bayes (GNB) classification process is the imbalance in class distribution within the dataset. Data imbalance occurs in datasets with unequal class ratios/cases [15],[16]. This class imbalance results in the classification tendency to favor the majority class while neglecting the minority class [17],[18]. The Synthetic Minority Over-sampling Technique-Nominal (SMOTE-N) method, an extension of SMOTE, is designed to address class imbalances in nominal feature datasets [19].

Based on the above exposition, this research focuses on the implementation of PPCA for missing data imputation and SMOTE-N class balancing using Naïve Bayes classification. Classification of Hepatitis using Naive Bayes with an imbalanced number of classes resulted in an AUC value of 0.816, as reported in the

study by Amrin et al. [3]. This study was conducted to examine the impact of classification evaluation by adding data preprocessing, handling missing data using PPCA imputation, and balancing the number of classes using the SMOTE-N method before performing the classification process. The aim is to compare the classification evaluation of hepatitis disease using Naive Bayes with the PPCA imputation approach and SMOTE-N class balancing.

II. METHOD

The research workflow involves comparing the classification results of Naïve Bayes. The first scenario begins with data collection, followed by imputing missing data using PPCA. The imputed data is then evaluated using Naive Bayes classification. The second scenario also starts with data collection and PPCA-based imputation. Subsequently, the imputed data is balanced using SMOTE-N before being evaluated with Naive Bayes classification. The proposed scenario is illustrated in Fig. 1.

A. Data

The data used in this research is the hepatitis dataset obtained from the UCI Machine Learning Repository, which can be accessed at https://archive.ics.uci.edu/dataset/46/hepatitis. The hepatitis dataset consists of 155 entries and 19 features, but there are 15 features with missing values. The data is divided into 2 classes with categories "die" (1) and "live" (2). The "die" class comprises 32 entries, while the "live" class comprises 123 entries. This data is the same as that used in the study [3].

The issue with this dataset is that some data are missing in several features. The missing data problem may arise due to errors during data collection, where complete results were not obtained, or during data input, where the person responsible for collecting the data did not have complete information. In this study, the data obtained from the UCI Repository is used in an incomplete state, and the number of data entries in each class is imbalanced. The dataset can be seen in Table I, where "?" indicates missing data in the dataset used.

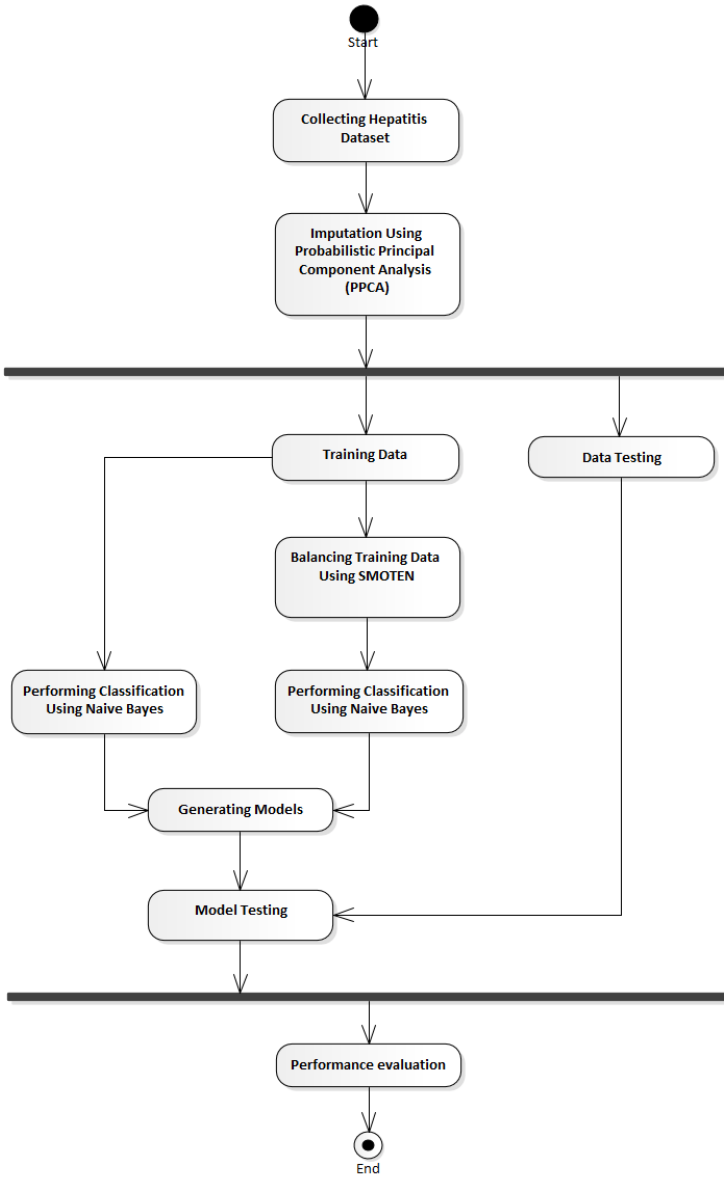

Fig. 1 The flow of research procedure

TABLE I
HEPATITIS DATASET

No.	Fitur								
	age	**sex**	**steroid**	**……**	**……**	**……**	**albumin**	**protime**	**histology**
1.	39	1	1	……	……	……	4	?	1
2.	50	2	1	……	……	……	3.5	?	1
3.	50	1	2	……	……	……	4	?	1
……	….	….	….	……	……	……	……	……	……
154.	53	2	1	……	……	……	4.1	48	2
155.	43	1	2	……	……	……	3.1	42	2

The number of missing data in each feature can be seen in Table II.

TABLE II
THE NUMBER OF MISSING DATA IN EACH FEATURE

No.	Feature Number	The Number of Missing Data
1.	age	0
2.	sex	0
3.	steroid	1
4.	antivirals	0
5.	fatigue	1
6.	malaise	1
7.	anorexia	1
8.	liver_big	10
9.	liver_firm	11
10.	spleen_palpable	5
11.	spiders	5
12.	ascites	5
13.	varices	5
14.	bilirubin	6
15.	alk_phosphate	29
16.	sgot	4
17.	albumin	16
18.	protime	67
19.	histology	0

B. Probabilistic Principal Componen Analysist (PPCA)

Probabilistic Principal Component Analysis (PPCA) is an imputation method that utilizes the EM algorithm (Expectation-Maximization) to iteratively compute Maximum Likelihood Estimates (MLE) from incomplete data sets. Principal Component Analysis (PCA) is a method used for dimensionality reduction. PCA reduces the dimensionality of data by linearly representing original data variables in a lower-dimensional space. MLE is one method used to estimate unknown/missing parameters from data. Each iteration of the EM algorithm during the data imputation process consists of two steps, including the Expectation (E) step and the Maximization (M) step [14]. The weakness of PCA lies in its ability to interpret probabilities or read the likelihood generated from existing PCA algorithms [20].

The imputation process in PPCA involves several steps. In the first step, the PPCA model is initialized, including initializing model parameters such as the number of principal components, weight (W) initialization, and observation covariant value (C) initialization. Then, in the second step, the Expectation Step (E-Step) values are computed, which involve calculating the expected values of hidden variables using the existing model. This includes calculating the expected values of hidden components based on the available observational data. This process can be performed using (1).

$$C = WW^T + \sigma^2 I \quad (1)$$

The value of C or observation covariant is obtained from the expected values in WW^T of the hidden variables estimated based on observational data, weight matrix (W) and noise variance (σ^2) on the identity matrix I (identity matrix *dxd*) where d is the dimension of the hidden variables. Next, in the third step, the Maximization Step (M-Step) involves updating the model parameters (mean (μ), weight matrix, and noise variance) using the expected values computed in the previous step. To calculate the estimated mean, eq. (2) is used, the updated weight matrix based on the mean estimate is calculated using (3), and the value of the noise variance in observational data is determined in (4).

$$S = \frac{1}{N}\sum_{n=1}^{N}(t_n - \mu)\,(t_n - \mu)^T \quad (4)$$

The value of S is the sample or estimated mean of the covariance matrix based on the observed values (t_n).

$$W = SW\,(\sigma^2 + M^{-1}W^T SW)^{-1} \quad (3)$$

W is the weight matrix updated based on the estimated mean (S), observed data, expected values, or latent variables.

$$\sigma^2 = \frac{1}{d} tr(S - SWM^{-1}W^T) \quad (4)$$

The value of σ^2 is the estimated variance of the noise in the observed data. The E-Step and M-Step are iteratively repeated until convergence is achieved or until the maximum number of iterations is reached. These iterations aim to update the model parameters (mean, weight matrix, and noise variance) so that the PPCA model can provide better estimations for the missing data [21],[22].

C. Synthetic Minority Over-sampling Technique-Nominal (SMOTE-N)

Synthetic Minority Over-sampling Technique-Nominal (SMOTE-N) is an extension of SMOTE designed specifically to address class imbalance in nominal feature sets. SMOTE-N is an oversampling algorithm that improves upon the Random Over Sampling (ROS) algorithm to generate synthetic instances. It evolves from SMOTE by employing an oversampling approach to the minority class and mitigating overfitting caused by merely duplicating

instances. Instead, SMOTE-N generates synthetic instances based on data similarity using the K-Nearest Neighbor (KNN) technique. Synthetic instances are created for the minority class in the dataset, aiming to balance the class ratio to a one-to-one ratio or approach it, thereby improving learning outcomes. The calculation of SMOTE-N for nearest neighbors is performed using a modified version of the Value Difference Metric (VDM) introduced by Cost and Salzberg. The VDM measures the distance between feature values in feature vectors within the context of nominal data. The VDM matrix defines the distance between corresponding feature values for the generated feature vectors [19]. The calculation of nearest neighbors for the minority class nominal features in SMOTE-N is conducted using the value difference metric (VDM) with (5).

$$\Delta(X, Y) = w_x w_y \sum_{i=1}^{N} \delta(xi, yi)^r \quad (5)$$

The $\Delta(X, Y)$ n equation (5) represents the distance between two data instances X and Y, where wxwy is the weight value for each value found in the nominal feature (N), and $\delta(xi, yi)^r$ is the difference value between the nominal feature values x_i and y_i with r being the parameter used to control the influence of each value. The distance dist between two corresponding feature values is defined in (6) [15], [19].

$$\Delta(F_1, F_0) = \sum \frac{n}{i=1} \left| \frac{C_{1i}}{C_1} - \frac{C_{0i}}{C_0} \right| k \quad (6)$$

F_1 and F_0 represent two corresponding feature values. C_1 is the total number of occurrences of the feature value F_1, and C_{1^i} is the number of occurrences of the feature value F_1 for class i, the same applies to C_{0^i} and C_0. k is a constant, typically with a value of 1. The equation above is used to compute the matrix of difference values for each nominal feature provided in the feature vector. This approach provides a more accurate and precise geometric distance in the context of nominal data [15], [19].

D. Naïve Bayes Gausian

One method in data classification is Naive Bayes Classification. Naive Bayes Classification is a machine learning method that utilizes probability and statistics calculations, which predict the probability of future data based on previous data, and forecast future chances based on known experiences, known as the Bayes theorem [23],[24]. Naive Bayes Classification has similar classification capabilities to decision trees and neural networks. This classification method has been proven to have high accuracy and speed when applied to large datasets [5]. The difference between Naive Bayes and Gaussian Naive Bayes lies in the assumed probability distribution for the features in the data. The Gaussian Naive Bayes method assumes that each feature in the data has a Gaussian (normal) probability distribution. This normal distribution is used to represent continuous and stable data where most observations are centered around the mean value, making it suitable for use when the features in the data are numerical variables [25] can be seen in (7).

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \quad (7)$$

The parameters σ_y and μ_y are estimates used in maximum likelihood estimation. $P(x_i \mid y)$ represents the conditional probability of feature x_i given class or label y. Then the formula $\sqrt{2\pi\sigma_y^2}$ is a normalization factor in the Gaussian distribution, where σ_y^2 is the variance of class or label y, ensuring that the computed probabilities are standardized. The function $exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$ is used to measure how close feature x_i is to the mean μ_y of class or label y. The smaller the distance between x_i and μ_y the higher the probability.

E. Area Under The Curve

The Area Under the Curve (AUC) represents the area under a curve plotted with sensitivity and specificity values. The AUC value always ranges between 0 and 1. AUC is calculated based on the combined area of trapezoids formed by sensitivity and specificity points, where a value of 1 indicates perfect model performance (perfectly separates positive and negative classes). The higher the AUC value, the better the performance of the classification model in separating positive and negative classes. If the AUC approaches 1, the model has excellent ability to distinguish between positive and negative classes. If the AUC approaches 0.5, the model performs similarly to random decision-making [26].

III. RESULT AND DISCUSSION

A. Imputation PPCA and Gaussian Naïve Bayes Classification

The Naïve Bayes classification process on the Hepatitis data with PPCA imputation approach can be observed in Table III. In Table I, several data points marked with "?" indicate missing values. In Table III, these previously marked "?" data points have been imputed with numbers obtained from PPCA predictions. The imputed data is then evaluated using the Naive Bayes classification.

TABLE III
RESULT OF PPCA IMPUTATION APPLICATION

No.	Feature								
	age	**sex**	**steroid**	**……**	**……**	**……**	**albumin**	**protime**	**histology**
1.	39	1	1	……	……	……	4	66.00545751	1
2.	50	2	1	……	……	……	3.5	60.00070197	1
3.	50	1	2	……	……	……	4	64.30442859	1
……	….	….	….	……	……	……	……	….	……
154.	53	2	1	……	……	……	4.1	48	2
155.	43	1	2	……	……	……	3.1	42	2

Comparison of the training and testing data used in hepatitis disease classification using Naive Bayes can be seen in Table IV.

Fig. 2 shows the comparison of AUC values between training and testing data. The highest AUC value is obtained with a split of 80:20 for training and testing data, achieving an AUC value of 0.833.

B. Imputation PPCA and Class Balancing SMOTE-N In Naïve Bayes Classification

This scenario is implemented on the same data, namely the PPCA-imputed data as shown in Table III. The PPCA-imputed data in this scenario is still unbalanced, with two classes having different numbers of data. The "live" class contains 123 data points, while the "die" class contains 32 data points, resulting in an imbalance ratio of 3.84 for the hepatitis data. This means that the "live" class has approximately 3.84 times more data than the "die" class. Generally, if the imbalance ratio between the majority and minority classes is greater than 1.5 or 2, class balancing is performed. The comparison of the training data before and after the class balancing process using SMOTE-N can be seen in Table V.

The class balancing process using SMOTE-N is only performed on the training data, while the testing data remains unbalanced, as in the case of the 90:10 split for training and testing data. Therefore, 90% of the training data undergo SMOTE-N class balancing, while the remaining 10% of the testing data remains unbalanced. Before SMOTE-N balancing, there were 139 data points, and after balancing, there are 222 data points. The SMOTE-N parameter used in this study employs the sampling strategy parameter "minority," which indicates that only the minority class will be oversampled. The value of k-neighbors is set to 9, which specifies the number of nearest neighbors used to generate synthetic samples. The reason for choosing k = 9 is based on previous experiments, particularly the study [19], which demonstrated that 9 is the optimal value to provide a good balance, yielding sufficient variation in synthetic data without making it too similar to the original samples. The calculation of SMOTE-N for the nearest neighbors is performed using a modified method of the Value Difference Metric (VDM). The percentage of oversampling in the training and testing data split (90:10) based on data before and after SMOTE-N is 59.70%.

The data distribution in this scenario follows the same pattern as the previous scenario, as shown in Table IV. The results of implementing SMOTE-N can be observed in Table VI.

Fig. 3 shows that the highest AUC value was obtained in the implementation of PPCA imputation with SMOTE-N class balancing using Naive Bayes Classification with a data training and testing split of 90:10, achieving 0.875. Based on the obtained AUC values, it indicates that SMOTE-N significantly influences Naïve Bayes classification.

TABLE IV
COMPARISON OF TRAINING AND TESTING DATA

Training Data	Testing Data
70%	30%
80%	20%
90%	10%

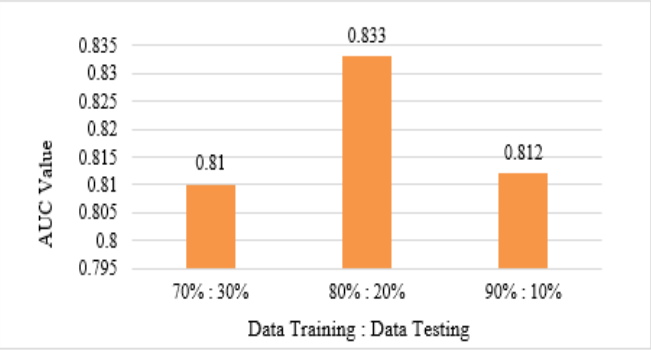

Fig. 2 The AUC value for Naïve Bayes classification using PPCA imputation

TABLE V
DATA COMPARISON SMOTE-N

Training Data / Class	Preliminary Data			Training Data SMOTE-N		
	70%	80%	90%	70%	80%	90%
Die	23	25	28	85	99	111
Live	85	99	111	85	99	111

TABLE VI
DATA AFTER PPCA IMPUTATION AND SMOTE-N CLASS BALANCING

No.	Feature								
	age	sex	steroid	……	……	……	albumin	protime	histology
1	39	1	1	…..	…..	…..	3.8	40	1
2	50	2	1	…..	…..	…..	3.4	41	2
3	50	1	2	…..	…..	…..	3.9	62	2
….	….	….	….	…..	…..	…..	….	….	….
219	30	1	1	…..	…..	…..	2.4	31	2
220	30	1	1	…..	…..	…..	2.4	31	2
221	31	1	1	…..	…..	…..	3.3	31	2

Fig. 4 illustrates the comparison results of two scenarios with the same data training and testing splits. For the 70:30 data training and testing split, the first model yielded an AUC of 0.810, while the second model achieved 0.817. In this comparison, the second model slightly outperformed the first. For the 80:20 data training and testing split, the first model resulted in an AUC of 0.833, whereas the second model obtained 0.830. Here, the first model had a slightly higher AUC than the second. Lastly, for the 90:10 data training and testing split, the first model attained an AUC of 0.812, while the second model reached 0.875. In this comparison, the second model had a higher AUC compared to the first. Therefore, the highest AUC value was achieved by implementing PPCA imputation, balancing classes using SMOTE-N, and Naive Bayes classification.

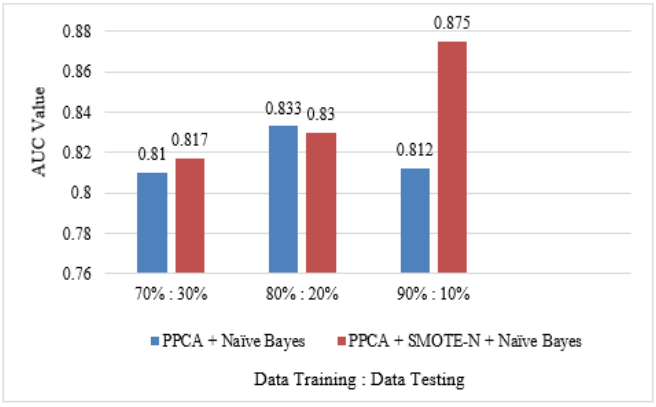

Fig. 4 Comparison result AUC Naïve Bayes classification

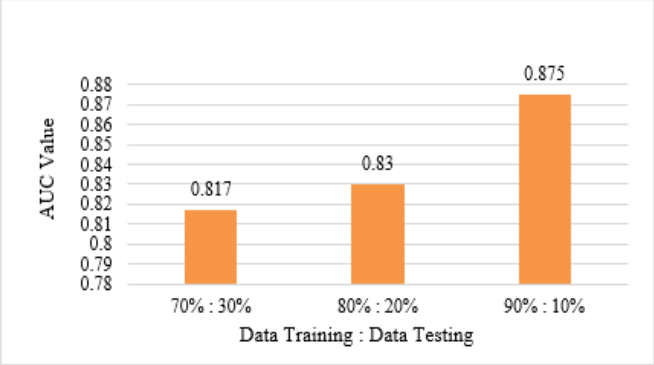

Fig. 3 AUC results for Naïve Bayes classification using PPCA and SMOTE-N

The highest AUC value achieved with the PPCA imputation approach using Naïve Bayes classification was obtained with an 80:20 split of training and testing data, resulting in 0.833. Meanwhile, the highest AUC value obtained with the PPCA imputation approach balanced with SMOTE-N using Naïve Bayes classification was achieved with a 90:10 split of training and testing data, resulting in 0.875. Upon further examination, it is observed that the AUC values obtained are not significantly different. Thus, it can be concluded that both research scenarios yield good results. The second scenario demonstrates an improvement in AUC

when incorporating the PPCA imputation and SMOTE-N class balancing approaches.

The data imputation process using PPCA aligns with research conducted by Hedge et al [14] which compared the results of MICE and PPCA imputation methods. Additionally, it is consistent with the work of Huang et al [22] who employed PPCA for imputation. The class balancing capability of SMOTE-N is also in line with studies by Wijayanti et al [19] and Kurniawati [15] ocusing on class balancing. Their research indicates that the advantages of the SMOTE-N method generally include avoiding information loss, preventing overfitting, and enhancing the prediction performance of minority classes, thereby yielding better performance after incorporating the SMOTE-N class balancing method. This study also aligns with the research by Husniah et al [7] , which implemented Naïve Bayes for predicting Hepatitis disease. Furthermore, the study by Febrian et al [8] focusing on diabetes prediction using supervised learning methods, demonstrated that Naive Bayes achieved high AUC values.

The average results obtained in both scenarios are as follows: in the first scenario, the implementation of PPCA imputation with Naïve Bayes classification yielded an average AUC value of 0.818, while in the second scenario, the application of PPCA imputation combined with SMOTE-N class balancing using Naïve Bayes classification resulted in a higher average AUC value of 0.84. The results of the second scenario are influenced by the combination of PPCA imputation and SMOTE-N. PPCA imputation serves as a method for handling missing data, while SMOTE-N functions to augment prediction data for minority classes during the training data process by maximizing the training process of the classification model. This leads to improved model performance during testing. In contrast, when classification is performed on data with missing values and imbalanced classes, it can adversely affect model training, resulting in suboptimal performance during testing.

IV. CONCLUSION

This study compared the classification evaluation of hepatitis disease using Naive Bayes with the PPCA imputation approach and SMOTE-N class balancing. The results obtained in the first scenario yielded the highest AUC value of 0.833 with an 80:20 split of training and testing data. Furthermore, in the second scenario, the highest AUC value of 0.875 was achieved with a 90:10 split of training and testing data. Based on these findings, it can be concluded that the PPCA imputation approach and SMOTE-N class balancing have a significant impact on Naïve Bayes classification. Additionally, this research demonstrates that the PPCA imputation method, SMOTE-N class balancing, and Naïve Bayes classification can be successfully implemented in hepatitis data. This underscores the potential of machine learning in healthcare, playing a crucial role in data identification processes.

ACKNOWLEDGEMENT

The author would like to express gratitude to Lambung Mangkurat University, Faculty of Mathematics and Natural Sciences, for their support during the research process.

REFERENCES

[1] H. P. Sari, D. Indriastuti, M. Asrul, and Elyasari, “Perbedaan Pengetahuan Pre Dan Post Pendidikan Kesehatan Pada,” *J. Keperawatan*, vol. 2, no. 3, pp. 9–16, 2019.

[2] A. T. Jalil, S. H. Dilfy, A. Karevskiy, and N. N. Mubark, “Viral hepatitis in Dhi-Qar province: demographics and hematological characteristics of patients,” *Int. J. Pharm. Res.*, vol. 12, no. 1, pp. 2081–2087, 2020, doi: 10.31838/ijpr/2020.12.01.326.

[3] Amrin and O. Pahlevi, “Implementasi Algoritma Klasifikasi Logistic Regression dan Naïve Bayes untuk Diagnosa Penyakit Hepatitis,” *J. Tek. Komput. AMIK BSI*, vol. 8, no. 2, pp. 174–180, 2022, doi: 10.31294/jtk.v4i2.

[4] H. Susana, N. Suarna, Fathurrohman, and Kaslani, “Penerapan Model Klasifikasi Metode Naive Bayes Terhadap Penggunaan Akses Internet,” *J. Ris. Sist. Inf. dan Teknol. Inf.*, vol. 4, no. 1, pp. 1–8, 2022, doi: 10.52005/jursistekni.v4i1.96.

[5] A. Haditsah, “Klasifikasi Masyarakat Miskin menggunakan Metode Naïve Bayes,” *Ilk. J. Ilm.*, vol. 10, no. 2, pp. 160–165, 2018.

[6] E. K. Ampomah, G. Nyame, Z. Qin, P. C. Addo, E. O. Gyamfi, and M. Gyan, “Stock market prediction with gaussian naïve bayes machine learning algorithm,” *Inform.*, vol. 45, no. 2, pp. 243–256, 2021, doi: 10.31449/inf.v45i2.3407.

[7] H. F. Husniah and T. Arifin, “Implementasi Algoritma Naïve Bayes Berbasis Particle Swarm Optimization Untuk Memprediksi Penyakit Hepatitis,” *J. Ilmu Komput.*, vol. 14, no. 2, pp. 37–49, 2019.

[8] M. E. Febrian, F. X. Ferdinan, G. P. Sendani, K. M. Suryanigrum, and R. Yunanda, “Diabetes prediction using supervised machine learning,” *Procedia Comput. Sci.*, vol. 216, no. 2022, pp. 21–30, 2022, doi: 10.1016/j.procs.2022.12.107.

[9] D. Derisma, “Perbandingan Kinerja Algoritma untuk Prediksi Penyakit Jantung dengan Teknik Data Mining,”

J. Appl. Informatics Comput., vol. 4, no. 1, pp. 84–88, 2020, doi: 10.30871/jaic.v4i1.2152.

[10] A. Ilham, "Hybrid metode boostrap Dan teknik imputasi pada metode C4-5 untuk prediksi penyakit ginjal kronis," *Statistika*, vol. 8, no. 1, pp. 43–51, 2020, [Online]. Available: https://jurnal.unimus.ac.id/index.php/statistik/article/view/5765.

[11] M. Alabadla *et al.*, "Systematic Review of Using Machine Learning in Imputing Missing Values," *IEEE Access*, vol. 10, pp. 44483–44502, 2022, doi: 10.1109/ACCESS.2022.3160841.

[12] P. Madley-Dowd, R. Hughes, K. Tilling, and J. Heron, "The proportion of missing data should not be used to guide decisions on multiple imputation," *J. Clin. Epidemiol.*, vol. 110, pp. 63–73, 2019, doi: 10.1016/j.jclinepi.2019.02.016.

[13] M. Lutfi and M. Hasyim, "Penanganan data missing value pada kualitas produksi jagung dengan menggunakan metode K-NN Imputation pada algoritma C4.5," *J. Resist.*, vol. 2, no. 2, 2019.

[14] H. Hegde, N. Shimpi, A. Panny, I. Glurich, P. Christie, and A. Acharya, "MICE vs PPCA: Missing data imputation in healthcare," *Informatics Med. Unlocked*, vol. 17, no. November, p. 100275, 2019, doi: 10.1016/j.imu.2019.100275.

[15] Y. E. Kurniawati, "Class Imbalanced Learning Menggunakan Algoritma Synthetic Minority Over-sampling Technique – Nominal (SMOTE-N) pada Dataset Tuberculosis Anak," *J. Buana Inform.*, vol. 10, no. 2, p. 134, 2019, doi: 10.24002/jbi.v10i2.2441.

[16] E. Erlin, Y. Desnelita, N. Nasution, L. Suryati, and F. Zoromi, "Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak seimbang," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 3, pp. 677–690, 2022, doi: 10.30812/matrik.v21i3.1726.

[17] F. H. Alfebi and M. D. Anasanti, "Improving Cardiovascular Disease Prediction by Integrating Imputation, Imbalance Resampling, and Feature Selection Techniques into Machine Learning Model," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 17, no. 1, p. 55, 2023, doi: 10.22146/ijccs.80214.

[18] M. P. Pulungan, A. Purnomo, and A. Kurniasih, "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Kepribadian MBTI Menggunakan Naive Bayes Classifier," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 7, pp. 1493–1502, 2023, doi: 10.25126/jtiik.1077989.

[19] N. P. Y. T. Wijayanti, E. N. Kencana, And I. W. Sumarjaya, "Smote: Potensi Dan Kekurangannya Pada Survei," *E-Jurnal Mat.*, vol. 10, no. 4, p. 235, 2021, doi: 10.24843/mtk.2021.v10.i04.p348.

[20] B. Wang, Z. Li, Z. Dai, N. Lawrence, and X. Yan, "A probabilistic principal component analysis-based approach in process monitoring and fault diagnosis with application in wastewater treatment plant," *Appl. Soft Comput. J.*, vol. 82, 2019, doi: 10.1016/j.asoc.2019.105527.

[21] C. Song, S. Yoon, and V. Pavlovic, "Fast ADMM algorithm for distributed optimization with adaptive penalty," *30th AAAI Conf. Artif. Intell. AAAI 2016*, pp. 753–759, 2016, doi: 10.1609/aaai.v30i1.10069.

[22] L. Huang, Z. Li, R. Luo, and R. Su, "Missing Traffic Data Imputation with a Linear Generative Model Based on Probabilistic Principal Component Analysis," *Sensors*, vol. 23, no. 1, 2023, doi: 10.3390/s23010204.

[23] A. Afandi, N. Noviana, and D. Nurdianah, "Naive Bayes Method and C4.5 in Classification of Birth Data," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 16, no. 4, p. 435, 2022, doi: 10.22146/ijccs.78198.

[24] F. Tempola, M. Muhammad, and A. Khairan, "Perbandingan Klasifikasi Antara KNN dan Naive Bayes pada Penentuan Status Gunung Berapi dengan K-Fold Cross Validation," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 5, pp. 577–584, 2018, doi: 10.25126/jtiik.201855983.

[25] N. F. Mustamin, F. Aziz, F. Firmansyah, and P. Ishak, "Classification Of Maternal Health Risk Using Three Models Naive Bayes Method," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 17, no. 4, p. 395, 2023, doi: 10.22146/ijccs.84242.

[26] L. Qadrini, A. Sepperwali, and A. Aina, "Decision Tree Dan Adaboost Pada Klasifikasi Penerima Program Bantuan Sosial," *J. Inov. Penelit.*, vol. 2, no. 7, pp. 1959–1966, 2021.