APPLICATION OF SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE (SMOTE) TO OUTLIER DATA FOR PROBABILISTIC NEURAL NETWORK (PNN)

Ramdan Hayati[1)]
Isran K. Hasan[2)]
Novianita Achmad[3)]
[1,2)] Statistics Study Program, Faculty of Mathematics and Natural Sciences, Universitas Negeri Gorontalo, Gorontalo, Indonesia
[3)] Mathematics Study Program, Faculty of Mathematics and Natural Sciences, Universitas Negeri Gorontalo, Gorontalo, Indonesia
e-mail: isran.hasan@ung.ac.id

ABSTRACT

One common model of Artificial Neural Network (ANN) used in classification tasks is the Probabilistic Neural Network (PNN). PNN is an algorithm that utilizes probability functions, eliminating the necessity for a large dataset during its development process. In this research, the best model parameters were initially determined using the sigma parameter and Kernel Density Estimation (KDE) function on a randomly sampled dataset employing the Stratified Random Sampling (SRS) method. The optimal sigma parameter obtained from this process is 0.075, with a Gaussian KDE function. The data used in this study is related to direct marketing campaigns (phone calls) from Portuguese banking institutions collected by S´ergio. Subsequently, PNN is applied to this dataset to determine its Accuracy and F1-Score values. The results indicate an accuracy rate of 87.117% and an F1-Score of 92.755%. Following this, Synthetic Minority Over-Sampling Technique (SMOTE) is applied to the dataset to balance the data. PNN is then implemented on the oversampled data, and in this phase, an evaluation of the Accuracy and F1-Score values is conducted, resulting in respective figures of 93.437% and 93.511%.

Keywords: probabilistic neural network, artificial neural network, gaussian kde function, synthetic minority over-sampling technique.

INTRODUCTION

Advancements in the field of biological neural networks have enabled researchers to create mathematical models that describe neurons, allowing for the simulation of neural network behavior. Artificial Neural Networks (ANNs) are simulations of biological nervous systems designed to comprehend patterns in data and simulate biological learning adaptation processes, albeit on a much simpler scale. According to (Wang, 2003), an Artificial Neural Network (ANN) consists of layers of input neurons (or nodes, units), one or two (or even three) layers of hidden neurons, and a final layer of output neurons. Equation 2.1 illustrates a typical architecture, where lines connecting neurons are also indicated. Each connection is associated with a numerical value known as the weight. One commonly used ANN model in classification tasks is the Probabilistic Neural Network (PNN). PNN is an algorithm that utilizes probability functions, eliminating the need for a large dataset during its development process. It possesses advantages in addressing several issues often encountered in backpropagation methods, such as lengthy training times, challenges in reaching global minimum values, and difficulties in designing network architectures. In PNN, learning falls into the category of supervised learning, obviating the need for repeated iterations to correct parameters used for data class identification. The mentioned parameters include the smoothing parameter (Adyati, Nasution, & Wahyuningsih, 2019)

According to (Specht, 1990) the Probabilistic Neural Network is a type of artificial neural network that integrates classification methods focused on probability with the concept of a neural network. In 1988, Donald F. Specht was one of the early developers of the PNN method. The Probabilistic Neural Network falls under the feedforward category in the types of neural networks.

In general, there are many situations where the number of occurrences in one class is significantly lower than the number of occurrences in another class. This phenomenon is known as class imbalance, which can lead to decreased performance of classification algorithms in various data mining applications, including medical pattern recognition, telecommunications management, bioinformatics, and text categorization. Many conventional models tend to assign data to the majority class and pay less attention to the minority class due to data imbalance. Therefore, preprocessing steps are required to address this class imbalance, and one oversampling technique that can be applied is the Synthetic Minority Oversampling Technique (SMOTE). By implementing SMOTE, the data distribution can be balanced by adding samples to the class with fewer samples (minority class) through synthetic data generated by SMOTE (Sutoyo & Fadlurrahman, 2020)

The Synthetic Minority Oversampling Technique (SMOTE) method is a technique employed to address imbalance between different classes through oversampling. The SMOTE approach is utilized to generate copies of minority data with the aim of achieving balance with the majority data. The SMOTE technique has the capability to alleviate overfitting issues commonly encountered in oversampling techniques (Shen, 2016)

Previous research related to classification and utilizing the PNN method was conducted by Er et al. (2021), who compared the PNN method with Multilayer Neural Network (MLNN) and Learning Vector Quantization (LVQ). The results obtained indicated that PNN 3 was the best classification with an accuracy of 96.30%, achieved through a 3-fold cross-validation. Furthermore, research on SMOTE was conducted by (Erlin, Desnelita, Nasution, Suryati, & Zoromi, 2022), who assessed the impact of using Synthetic Minority Oversampling Technique (SMOTE) on the Random Forest classification algorithm in the context of heart disease prediction. The findings revealed that SMOTE could reduce overfitting issues while improving the performance of the Random Forest model across all parameters. Another study on SMOTE was carried out by (Susanti, 2019), who combined the SMOTE algorithm with the Classifier Neural Network algorithm to address the issue of imbalanced class in the Scadi dataset. The research results showed that the use of SMOTE could enhance the performance of the neural network algorithm on the Scadi dataset, which was used to classify seven classes. The accuracy obtained was 90.47%, compared to the pre-SMOTE accuracy of only 80%.

METHOD

The stages of this research can be seen in Figure 1 below.

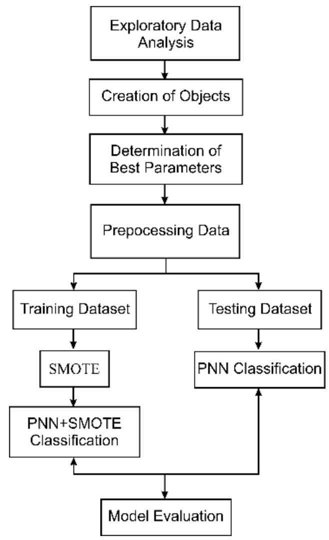

Figure 1. Research Stages

Data Explorations

Data explorations or Exploratory Data Analysis (EDA) is an approach in statistics and computer science aimed at detailing and analyzing patterns within data. With a focus on the initial understanding of dataset characteristics, EDA utilizes various graphical techniques and descriptive statistics to help identify relationships, anomalies, and interesting patterns within the data. Additionally, EDA assists researchers or analysts in forming initial hypotheses that can be further tested using more in-depth analysis methods. By delving into the distribution, relationships, and trends within the data, EDA paves the way for a better understanding of the phenomena contained in the dataset, guiding the decision-making process in subsequent stages of data analysis. In other words, EDA plays a crucial role in detailing the context and meaning behind each dataset, unlocking the potential for new insights and creative problem-solving.

Descriptive Analysis

Descriptive analysis in the research on the Probabilistic Neural Network (PNN) is conducted to provide a detailed overview of the characteristics and performance of the model. This analysis includes descriptive statistics related to key parameters in the PNN, such as accuracy rate, precision, recall, and F1-score (Azizah, 2021). Through graphical visualization and tables, descriptive analysis can also depict the distribution of PNN predictions on training and testing data. Furthermore, this analysis can detail the performance of PNN in handling tendencies of overfitting or underfitting. The results of the descriptive analysis in the PNN research offer an initial understanding of the model's reliability in the specific context of the used dataset, aiding researchers in evaluating and comprehending how effectively PNN can be applied to the researched problem (Er, 2011).

Probabilistic Neural Network

Probabilistic Neural Network is a type of artificial neural network that integrates a classification method focused on probability with the concept of neural networks. In 1988, Donald F. Specht was an early developer of the PNN method. The Probabilistic Neural Network is a part of the feedforward category in the types of neural networks (Utomo, Gumilang , & Ahmad, 2022)

PNN has several notable advantages, especially in addressing data with non-linear characteristics, training speed, generalization ability, managing data noise, and handling classification problems with multiple classes. Although there are other effective classification methods, PNN can be a very good choice, particularly when the data exhibits specific characteristics (Mohebali, 2020)

PNN Algorithm

The testing algorithm with the Probabilistic Neural Network is as follows (Nurbaiti, Setyaningsih, & Midyanti, 2017):

1. Input Layer

   The input contains testing data that will be used in the PNN calculations

2. Patter Layer

   The calculation of the distance between testing data and training data can be found in equation (1).

$$y(x) = e \, \frac{-(x - X)^2}{\sigma^2} \tag{1}$$

3. Summation Layer

   This calculation process involves computing the averages for each class, allowing the determination of the probability of a specific input belonging to its class, as stated in equation (2).

$$Summition\ Layer = \sum y(x) \tag{2}$$

4. Output Layer

   The result obtained from the PNN method is the maximum value among the other classes, which is calculated in the summation layer as explained in equation (3).

$$Output\ Layer = Max\ (gx) \tag{3}$$

Fungsi Kernel Density Estimation

According to Sarita et al. (2019), Kernel Density Estimation (KDE) is a non-parametric statistical method used to estimate the probability distribution of a continuous random variable. KDE smoothens or "softens" the data distribution, providing a continuous visual estimate of the distribution without assuming a specific distribution shape. For example, given a random sample of size n, namely X1, X2,

..., Xn originating from an unknown density function f, the kernel density function estimation is formulated as follows:

$$\hat{f}(x,h) = \frac{1}{nh}\sum_{i=1}^{n} K\,(\frac{x - X_i}{h}) \tag{4}$$

According to (Sarita, Setiawan, & Parhusip, 2019), several commonly used functions and forms of kernels include:

1. **Gaussian:** The most commonly used type of kernel function in density estimation. This function utilizes the normal distribution function as its kernel.

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\,(\frac{x^2}{2}) \tag{5}$$

2. **Triangular:** A type of kernel function in density estimation that uses a triangular function as its kernel.

$$K(x) = (1 - |x|)\ for\ |x| < 1 \tag{6}$$

3. **Epanechnikov**: A type of kernel function in density estimation that utilizes the Epanechnikov function as its kernel.

$$K(x) = \left(\frac{3}{4}\right)(1 - x^2) for\ |x| < 1 \tag{7}$$

4. **Laplacian**: A type of kernel function in density estimation that uses the Laplacian function as its kernel.

$$K(x) = e^{-}\frac{|x|}{\sigma} K(x) = (1 - |x|)\ for\ |x| < 1 \tag{8}$$

5. **Uniform (Box)** : Applying a constant weight to all values within a specific distance from the kernel center, and the weight becomes zero outside that distance.

$$K(x) = \left(\frac{1}{2}\right) for\ |x| < 1 \tag{9}$$

Synthetic Minority Oversampling Technique

According to (Chawla, Bowyer, Hall, & Kegelmeyer, 2002), the SMOTE (Synthetic Minority Over-sampling Technique) method is the most popular oversampling method due to its simplicity, computational efficiency, and superior performance. The SMOTE approach is a method used to achieve balance between different classes through oversampling. With SMOTE, data from the minority class is increased by generating duplicates, aligning it with the majority class's data volume. This approach has the ability to reduce the overfitting issues that often arise in oversampling techniques.

The SMOTE approach is a method used to achieve balance between different classes through oversampling. With SMOTE, data from the minority class is enhanced by generating duplicates, aligning it with the quantity of majority class data. This approach has the ability to alleviate the issue of overfitting that often arises in oversampling techniques (Shen, 2016)

Oversampling is applied to the training data. The SMOTE method is used to generate synthetic data in the minority class (in this context, class 1), so that the number of entities in the minority class becomes equivalent to the number of entities in the majority class (in this context, class 0). The goal is to achieve data balance in both classes. This addition of synthetic data applies only to the training data (Arifiyanti & Wahyuni, 2020).

RESULTS AND DISCUSSION

Data Explorations

This data is associated with a direct marketing campaign (phone calls) conducted by Portuguese banking institutions and collected by Sérgio (Moro, Cortez, & Rita, 2014). Based on its output, the data is categorized into two classes: "Yes" for customers subscribing to deposits and "No" for customers not subscribing to deposits. The data that has undergone preprocessing is then divided into two datasets: training data and test data. The training data is used for building the classification model. The data is split using the hold-out method with a training to test data ratio of 80:20. From this split, the amount of data is 45.211, as shown in the diagram below.

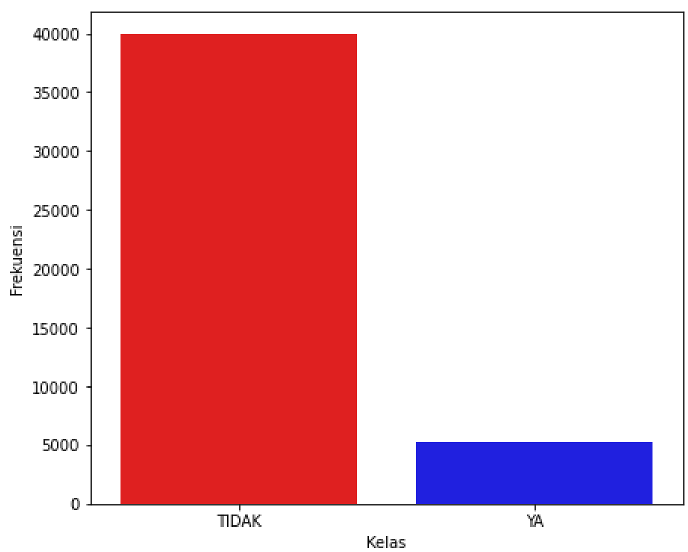

Figure 2. Frequency Diagram of Class

Based on Figure 1, it can be observed that the "NO" class of customers not subscribing to deposits dominates, amounting to 39,922 data out of the total Bank Marketing data training Portugal, which is 45,211. It can be stated that the data is imbalanced. Subsequently, an examination of outlier data is conducted.

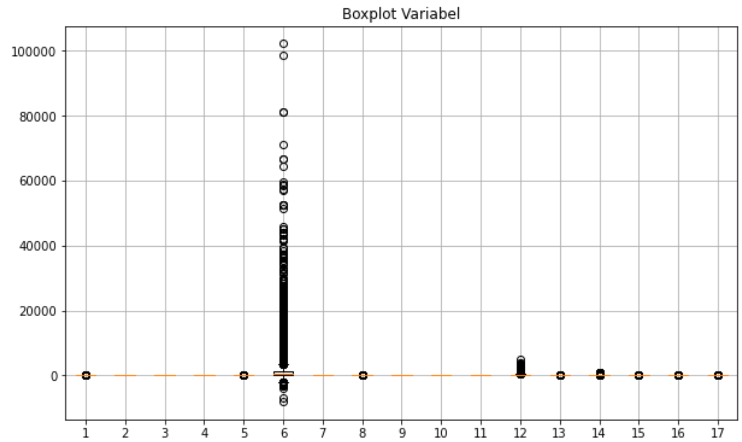

Figure 3. Outlier of Data

In Figure 2, it can be observed that there are outliers in the visualization of the data above. This indicates that the line in the middle of the box represents the median value of the data, while the points outside the whiskers indicate values considered as outliers. The image reflects the presence of some variables with outliers, such as variables X1, X6, X12, X13, X14, and X15.

Descriptive Analysis

The data used by the researcher is sourced from the UCI Repository, collected by Sérgio, related to direct marketing campaigns (phone calls) from Portuguese banking institutions. There are 16 features in the data, including Age (X1), Job (X2), Marital Status (X3), Education (X4), Default (X5), Balance (X6), Housing (X7), Loan (X8), Contact (X9), Day (X10), Month (X11), Duration (X12), Campaign (X13), P Day (X14), Previous (X15), P Outcome (X16). Researchers chose to utilize all attributes from the data sourced from the UCI Repository to gain a comprehensive understanding of the available data characteristics and to explore each attribute's contribution to the classification process. To examine the characteristics of the data, descriptive analysis is necessary, as shown in the following Table 1.

Table 1. Descriptive Analysis

Variabel	Count	Mean	Std	Min	25%	50%	75%	Max
Age	45211	40.93621	10.61876	18	33	39	48	95
Balance	45211	1362.272	3044.766	-8019	72	448	1428	102127
Day	45211	15.80642	8.322476	1	8	16	21	31
Duration	45211	258.1631	257.5278	0	103	180	319	4918
Campaign	45211	2.763841	3.098021	1	1	2	3	63
Pdays	45211	40.19783	100.1287	-1	-1	-1	-1	871
Previous	45211	0.580323	2.303441	0	0	0	0	275

It can be seen from Table 4.1 that the range of values for each of the 32 variables varies, such as in variables X1, X6, X12, and X14, which show high values compared to other variables. Therefore, data standardization is necessary to bring the values of each variable into the same range.

Standardization Techniques

Data standardization is carried out to ensure that the data has a consistent format and can be interpreted correctly. The data standardization process transforms the mean and standard deviation values of each feature to zero and one, respectively. This makes the data have a normal distribution, facilitating machine learning algorithms that are sensitive to scale in processing it. The results of the standardization process are presented in Table 2.

Table 2. Data After Standardization

No	Variable				
	X1	X2	X3	…	X16
1	1.607	-0.104	-0.276	…	0.445
2	0.289	1.424	1.368	…	0.445
3	-0.747	-0.715	-0.276	…	0.445
4	0.571	-1.021	-0.276	…	0.445
5	-0.747	2.035	1.368	…	0.445
6	-0.559	-0.104	-0.276	…	0.445
7	-1.218	-0.104	1.368	…	0.445
8	0.100	-0.715	-1.920	…	0.445
9	1.607	0.202	-0.276	…	0.445
10	0.194	1.424	1.368	…	0.445
11	0.006	-1.326	-1.920	…	0.445
12	-1.124	-1.326	1.368	…	0.445
13	1.136	1.424	-0.276	…	0.445
14	1.607	1.424	-0.276	…	0.445
15	1.513	0.813	-0.276	…	0.445
16	0.948	0.202	-0.276	…	0.445
17	0.383	-1.326	1.368	…	0.445
18	1.513	-1.021	-0.276	…	0.445
19	1.795	0.202	-0.276	…	0.445
…	…	…	…	…	…
45209	2.925	0.202	-0.276	…	-0.566
45210	1.513	-1.021	-0.276	…	0.445
45211	-0.371	-0.715	-0.276	…	-1.577

It can be observed from Table 2 that after standardization, the data is centered around 0. Values below 0 indicate that the first data point is smaller than the average, a value of 0.0 indicates that the second data point is equal to the average, and values above 0 indicate that the third data point is larger than the average. Next is the creation of the PNN Object.

Creating an Object for the PNN Model

At this stage, the PNN model object is created using the sigma parameter and Kernel Density Estimation (KDE) function to determine the best parameters. Sigma is a parameter that determines the size of the kernel in the kernel density estimation function. This parameter stores information about the kernel density estimation function, such as the data used to create the kernel density estimation function, the size of the kernel bandwidth, and the probability values of each point in the data space. The KDE function is a function used to estimate the probability distribution of data. This function can be used to estimate the probability distribution of data, which can be utilized for various purposes such as classification, regression, and clustering.

Determining the Best Parameters

The kernel density function can be used to provide a visual or numerical representation of the data distribution, even if the distribution is not explicitly known. The KDE functions used in this research include Gaussian, Laplacian, Triangular, Epanechnikov, and Uniform . This estimation focuses on the concept of weighting (kernel) around each data point taken from the sample observations. As for the sigma parameter (smoothing parameter) used, it includes values such as 0.010, 0.050, 0.075, 0.500, 0.800, 1.000, and 1.200 (Ramadhani, Ispriyanti, & Safitri, 2018).

Due to the very large amount of data, the determination of the Kernel Density Estimation Function is carried out by taking a 5% sample of the total data. This is done to address the complexity of calculations and allow efficiency in the analysis process. The sampling is performed using stratified random sampling to ensure that each significant subgroup in the population is proportionally represented in the sample, thus reducing potential sampling errors.

The next step is to apply the PNN model to the data training to determine the best parameters for the Kernel Density Estimation function. With the help of the Python program, the estimated kernel density results are obtained for each function, as shown in Table 3.

Table 3. Parameter Determination Output

Kernel	Smoothing Param	Accuracy	F1-Score
Gaussian	0.01	0.873894	0.93040293
Gaussian	0.05	0.876106	0.93253012
Gaussian	0.075	0.882743	0.93772033
Gaussian	0.5	0.845133	0.91025641
Gaussian	0.8	0.836283	0.90314136
Gaussian	1	0.838496	0.90382082
Gaussian	1.2	0.818584	0.89066667
Triangular	0.01	0.882743	0.93772033
Triangular	0.05	0.882743	0.93772033
Triangular	0.075	0.882743	0.93772033
Triangular	0.5	0.882743	0.93772033

Kernel	Smoothing Param	Accuracy	F1-Score
Triangular	0.8	0.884956	0.93882353
Triangular	1	0.878319	0.93506494
Triangular	1.2	0.869469	0.93001186
Epanechnikov	0.01	0.882743	0.93772033
Epanechnikov	0.05	0.882743	0.93772033
Epanechnikov	0.075	0.882743	0.93772033
Epanechnikov	0.5	0.882743	0.93772033
Epanechnikov	0.8	0.884956	0.93882353
Epanechnikov	1	0.876106	0.93380615
Epanechnikov	1.2	0.869469	0.93001186
uniform	0.01	0.882743	0.93772033
uniform	0.05	0.882743	0.93772033
uniform	0.075	0.882743	0.93772033
uniform	0.5	0.882743	0.93772033
uniform	0.8	0.884956	0.93882353
uniform	1	0.876106	0.93380615
uniform	1.2	0.869469	0.93001186
laplacian	0.01	0.882743	0.93772033
laplacian	0.05	0.882743	0.93772033
laplacian	0.075	0.882743	0.93772033
laplacian	0.5	0.882743	0.93772033
laplacian	0.8	0.884956	0.93882353
laplacian	1	0.876106	0.93380615
laplacian	1.2	0.869469	0.93001186

Based on Table 3, the result shows that the best kernel density estimation function is the Gaussian function with a sigma value of 0.075, yielding an accuracy of 88.27% and an F1-Score of 93.77%.

Preprocessing Data

The imbalanced data related to the classification of customers subscribing to deposits from Portuguese banking institutions consists of a total of 45.211 entries. The data that has undergone preprocessing is then divided into two datasets: training data and test data. The training data is used for building the classification model. The data is split using the hold-out method with a training to test data ratio of 80:20.

Probabilistic Neural Network Classification

After obtaining the best parameters in the application of stratified random sampling, which is using the Gaussian function with a Sigma value of 0.075, the next step is to apply these parameters to the PNN classification with the entire data.

Table 4. Results of PNN Application

Data	Sigma	Accuracy	F1-Score
Test	0.075	87,117%	92,755%
Train	0075	88,239%	93,452%

Based on Table 4, the accuracy and F-1 Score values obtained from the implementation of PNN on the testing data with the respective object are 0.8711710715 or 87.117% and 0.927554257 or 92.755%. For the training data, the values are 0.8823876901 or 88.239% and 0.9345210385 or 93.452%.

Synthetic Minority Oversampling Technique (SMOTE) Application

Imbalanced data handling in this study was carried out using the Synthetic Minority Oversampling Technique (SMOTE). The SMOTE method was applied to oversample the minority class in the training data, where the minority class in this study is the YES class. Synthetic data was generated for the YES class to make its quantity equal to the final proportion in each class. As seen in Figure 4 below, the YES class is obtained as 39,922, which is equivalent to the NO class.

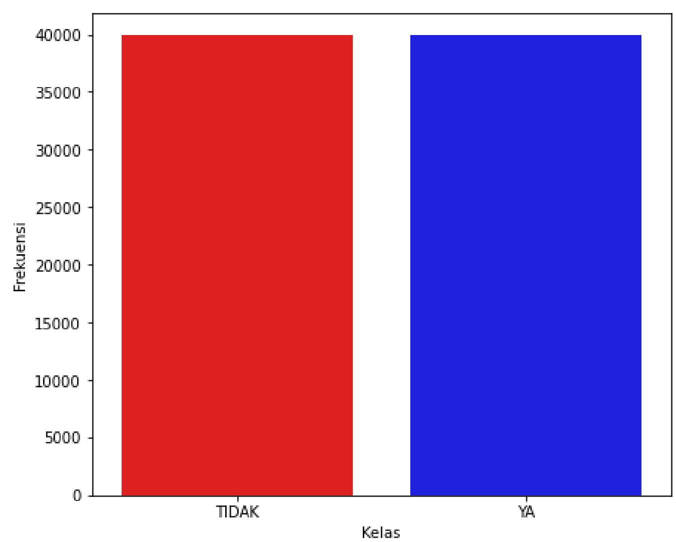

Figure 4. Training dataset after SMOTE

Based on table 5, the accuracy and F-1 Score values obtained from the application of SMOTE-generated on the testing data are 0.9343728473918 or 93.437% and 0.9351163942546 or 93.511%, respectively. For the training data, the values are 0.933279690211 or 93,328% and 0.937335472385 or 93.733%.

Table 5. Results of PNN+SMOTE Application

Data	Sigma	Accuracy	F1-Score
Test	0.075	93,437%	93,511%
Train	0.075	93,328%	93,733%

Comparison of Prediction Results between PNN and PNN+SMOTE

Table 6 explains the comparison of prediction results between PNN formed with and without SMOTE on the testing data. Based on the classification results in the table, it can be seen that the classification with SMOTE-treated data tends to produce better classification results with an accuracy value of 93.437% and an F-1 Score of 93.511%, compared to the classification results on data without SMOTE treatment with an accuracy of 87.117% and an F-1 Score of 92.755%.

Table 6. Prediction Results between PNN and PNN+SMOTE

Sigma	Accuracy	F1-Score
0.075	87.117%	92,755%
0.075	93,437%	93,511%

CONCLUSION

Based on the analysis and discussion conducted, the conclusions drawn from the PNN classification for imbalanced data and data indicating outliers using 16 independent variables (e.g., Age, Occupation, Marital Status, Education, Default, Balance, Housing, Loan, Contact, Day, Month, Duration, Campaign, Pdays, Previous, Poutcome) and 1 dependent variable (Deposit Category) are as follows the classification accuracy of PNN for imbalanced data is 87.117%, indicating a good performance in handling the original dataset. Experimenting with PNN classification on imbalanced data while applying SMOTE resulted in improved accuracy, achieving the best accuracy of 93.437%. It can be seen that there is an increase in accuracy by 6.76%. This suggests that the SMOTE technique effectively addresses overfitting issues, producing a more robust and stable model. In conclusion, the application of SMOTE, a Synthetic Minority Oversampling Technique, proves to be beneficial in enhancing the classification performance of PNN for imbalanced datasets. The resulting model demonstrates improved accuracy, indicating its effectiveness in handling imbalanced class distribution and potential outlier issues.

REFERENCE

Adyati, R. D., Nasution, Y. N., & Wahyuningsih, S. (2019). KLASIFIKASI PROBABILISTIC NEURAL NETWORK (PNN) PADA DATA DIAGNOSA PENYAKIT DEMAM BERDARAH DENGUE (DBD) TAHUN 2018. *Prosiding Seminar Nasional Matematika Dan Statistika*, *1*, pp. 15-21. Retrieved from https://jurnal.fmipa.unmul.ac.id/index.php/SNMSA/article/view/521

Arifiyanti, A. A., & Wahyuni, E. d. (2020). SMOTE: METODE PENYEIMBANG KELAS PADA KLASIFIKASI DATA MINING. *Jurnal Teknologi Informasi dan Komunikasi*, 34-39.

Azizah, P. D. (2021). Penerapan Probabilistic Neural Network pada Klasifikasi Berat Bayi. *Journal Riset Statistika, 1*(2), 152-159. doi:doi:10.29313/jrs.v1i2.524.

Chawla, N. V., Bowyer, K., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research, 16*(1), 321-357.

Er, O. (2011). An approach based on probabilistic neural network for diagnosis of Mesothelioma's disease. *Computers & Electrical Engineering*, 75-81.

Erlin, E., Desnelita, Y., Nasution, N., Suryati, L., & Zoromi, F. (2022). Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak seimbang. *Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer, 21* (3), 677-690. Retrieved from https://doi.org/https://doi.org/10.30812/matrik.v21i3.1726

Mohebali, B. T.-B. (2020). Probabilistic neural networks: a brief overview of theory, implementation, and application. In P. Samui, D. T. Bui, S. Chakraborty, & R. C. (Eds.), *Handbook of Probabilistic Models* (pp. 347-367). Retrieved from https://doi.org/10.1016/B978-0-12-816514-0.00014-X.

Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems, 62*, 22-31. doi:https://doi.org/10.1016/j.dss.2014.03.001

Nurbaiti, Setyaningsih, F. A., & Midyanti, D. (2017). IDENTIFIKASI BIBIT PADA TANAMAN LAHAN GAMBUT BERDASARKAN BENTUK DAUN MENGGUNAKAN METODE PROBABILISTIK NEURAL NETWORK (PNN)BERBASIS WEBSITE (UMUR BIBIT 2 BULAN-1 TAHUN). *CODING: Jurnal Komputer dan Aplikasi, 5*(1), 14-22. doi:DOI: http://dx.doi.org/10.26418/coding.v5i1.19170

Ramadhani, P., Ispriyanti, D., & Safitri, D. (2018). KAPABILITAS PROSES DENGAN ESTIMASI FUNGSI DENSITAS KERNEL PADA PRODUKSI DENIM DI PT APAC INTI CORPORA. *JURNAL GAUSSIAN, 7*(1), 326-336.

Sarita, F. T., Setiawan, A., & Parhusip, H. (2019). Analisis Indeks Pembangunan Manusia (IPM) Kabupaten/Kota di Provinsi Maluku Utara Menggunakan Indeks Geary C Berdasarkan Resampling Estimasi Densitas Kernel. *JuTISI : Jurnal Teknik Informatika dan Sistem Informasi, 5*(1), 62-72. doi:https://doi.org/10.28932/jutisi.v5i1.1582

Shen, L. L. (2016). *Relay Backpropagation For Effective Learning Of Deep Convolutional Neural Networks.* (B. M. In Leibe, Ed.) Cham: Springer International.

Specht, D. F. (1990). Probabilistic Neural Networks. *Neural Networks, 3*(1), 109-118. doi:https://doi.org/10.1016/0893-6080(90)90049-Q

Susanti, S. (2019). Klasifikasi Kemampuan Perawatan Diri Anak dengan Disabilitas Menggunakan SMOTE Berbasis Neural Network. *JURNAL INFORMATIKA, 6*(2), 175-184.

Sutoyo, E., & Fadlurrahman, M. (2020). Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network. *JEPIN (Jurnal Edukasi dan Penelitian Informatika), 6*(3), 379-385. doi:http://dx.doi.org/10.26418/jp.v6i3.42896

Utomo, A., Gumilang , M., & Ahmad, A. (2022). Agricultural Commodity Sales Recommendation System For Farmers Based on Geographic Information Systems and Price Forecasting Using Probabilistic Neural Network Algorithm. *The 4th International Conference on Food and Agriculture* (pp. 1-9). Jember: IOP Conference Series Earth and Environmental Science.

Wang, S.-C. (2003). *Artificial Neural Network In: Interdisciplinary Computing in Java Programming.* Boston: Springer, Boston, MA. doi: https://doi.org/10.1007/978-1-4615-0377-4_5.