JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 9 Number 1 | June 2024: 138-147
DOI: 10.15575/join.v9i1.1293

Data Balancing Techniques Using the PCA-KMeans and ADASYN for Possible Stroke Disease Cases

Uung Ungkawa[1], Muhammad Avilla Rafi[2]
[1,2]Department of Informatics, Institut Teknologi Nasional Bandung, Indonesia

Article Info

Article history:

Received January 03, 2023
Revised October 27, 2023
Accepted April 21, 2024
Available Online June 30, 2024

Keywords:

ADASYN
Imbalanced Data
Machine Learning
PCA-KMeans
Stroke

ABSTRACT

Imbalanced data happens when the distribution of classes is not equal between positive and negative classes. In healthcare, the majority class typically consists of healthy patient data, while the minority class contains sick patient data. This condition can cause the minority class prediction to be wrong because the model tends to predict the majority class. In this study, we use a deep neural network algorithm with focal loss that can deal with class imbalance during training. To balance the data, we use the PCA-KMeans combination model to shrink the dataset and the ADASYN model to give the minority class more samples than it needs. In this study, the research problem is how well the two techniques can improve model performance, especially in minority case classification. The mild model is the best without data balancing, resulting in an accuracy value of 84%. The class 0 F1-score has a value of 86%, whereas the class 1 F1-score has a value of 82%. The moderate model is the best model in the case study of PCA-KMeans balancing data, resulting in an accuracy value of 89%; the class 0 F1-score is 91%; and the class 1 F1-score is 85%. The extreme model is the best model in the ADASYN data balancing case study, resulting in an accuracy value of 95%; the value in class 0 gets a F1-score of 96%, while the value in class 1 gets a F1-score of 96%. Of the three test models, the best model is obtained using ADASYN extreme data balancing with an accuracy value of 95%, the value in class 0 with a F1- score of 93%.

Corresponding Author:

Uung Ungkawa
Informatics Study Program, Faculty of Industry Technology, Institut Teknologi Nasional Bandung
Jl. PH. H. Mustofa No.23, Bandung 40124, West Java, Indonesia
Email:uung@itenas.ac.id

1. INTRODUCTION

Imbalanced data is a condition where the distribution of data class ratios is not equal between positive and negative classes [1]. Various sectors, including the health field, often encounter imbalanced data. In the health sector, normal or healthy patient data typically belong to the majority class, whereas abnormal or sick patients fall into the minority class. This condition can cause the minority class prediction to be incorrect because the model tends to predict the majority class [2].

There are 3 levels of imbalance, namely mild with the proportion of minority classes ranging from 20–40% of the dataset, moderate with the proportion of minority classes ranging from 1-20% of the dataset, and extreme with the proportion of minority classes <1% of the dataset [3]. Generally, sampling methods or balancing the distribution of classes in a dataset can overcome data imbalances. We divide sampling techniques into two categories: oversampling and undersampling [4].

If we carry out the data balancing process, classification methods can generally obtain optimal results. However, there are some classification methods that have a function to overcome data imbalances, such as Deep Neural Networks (DNN) with focal loss functions. Deep Neural Network (DNN) with focal loss is a deep learning approach that addresses the issue of unbalanced classes in the dataset.

Focal loss will focus and improve the model's accuracy and performance in predicting data from minority classes [5].

Jing and Yuru conducted a machine learning performance analysis to predict strokes. Based on the Imbalanced Medical Dataset, Jing and Yuru discuss the extreme data imbalance that results in 3806 samples, of which less than 5% are stroke patients after the data cleaning process. This study uses four models for detecting stroke. The first model uses the SMOTE technique for oversampling minority classes. The results show improved performance on several machine learning algorithms, including the Logistic Regression (LR) algorithm, which increased the F1-score from 0.00 to 0.28 and the AUC from 0.50 to 0.71. The second model used is the Weighted-Voting Classifier which is a combined ensemble method of several machine learning algorithms that results in better performance compared to other methods. The third model uses Deep Neural Network (DNN) with Focal Loss. An approach that uses a modified loss function to handle class imbalance. The results show almost as good performance as using LR and SMOTE, with an f1-score of 0.31 and an AUC of 0.72. The last model has the best performance among the four, which is the combined undersampling with PCA-KMeans and DNN-Focal Loss. This method successfully reduces the class imbalance in the dataset and increases its size. The results show that the F1-score reached 0.77 and the AUC reached 0.90 after 200 epochs [6].

In the research entitled Comparative Analysis of ADASYN-SVM and SMOTE-SVM Methods on the Detection of Type 2 Diabetes Mellitus, the journal discusses data imbalance in a case study of diabetes mellitus with 630 rows with 9 features in the dataset, of which 290 are diabetic patients and 340 are non-diabetic patients. Due to data imbalance, researchers used the SMOTE and ADASYN methods to balance the amount of diabetic and non-diabetic data, and used SVM as a classifier. This study produced 2 models, namely ADASYN-SVM with an accuracy of 87.3% and SMOTE-SVM with an accuracy of 85.4% [7].

Based on the above explanation, we conduct comparative analysis on the level of data imbalance using the deep neural network algorithm method with a focal loss function that has the ability to handle class imbalance during training. The balancing techniques use a combination of PCA- KMeans to reduce the dimensions of the majority class [8] and Adaptive Synthetic Sampling (ADASYN) which oversamples the minority class [9]. So as to result in the best model of data balancing techniques with a balanced and optimal classification performance, especially in the minority class, that will be applied to the deep neural network algorithm to detect the possibility of stroke disease.

2. METHOD

An imbalanced dataset can cause the minority class prediction to be wrong because the model tends to predict the majority class [2]. To alleviate this problem, we propose to balance the dataset by using PCA-KMeans to reduce the size of the data and ADASYN to generate syntetic data.

There are three stages in this research: input, processing, and output. The dataset will go through several preprocessing steps, namely label encoding [10], one hot encoding [10], outlier removal [11], [12], missing values [13], and data normalization [14].

The next step involves processing the dataset for data balancing, utilizing the PCA-KMeans and ADASYN techniques to produce balanced data models. The data processing also splits the data for classification. We will divide the split data into training and testing data at a ratio of 80:20. The process produces three models: models without data balancing, modes with PCA-KMeans balancing, and models with ADASYN data balancing. All three models will be used to train DNN with focal loss as a classifier. The next step is to evaluate the performance of the three models and compare them to find the best model for each case study. As shown in Figure 1, the following is the research process.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

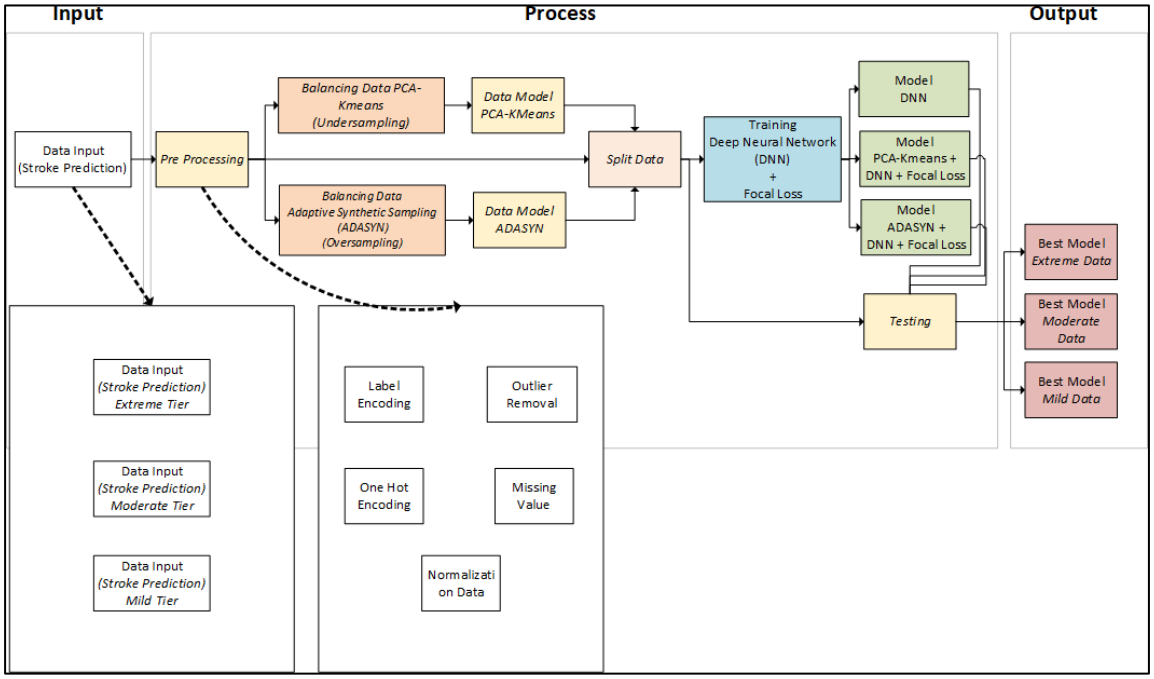

Figure 1. Research Method

2.1. Dataset Description

The Kaggle website [15] provided the dataset for this study, which predicts a patient's likelihood of having a stroke. with 11 features and 1 target label, for a total of 5110 data samples, which are divided into two classification classes, namely, Patients Not Likely to Have a Stroke and Patients Likely to Have a Stroke, as shown in Table 1.

Table 1. Stroke Dataset

Class	Description	Total Data
0	Patient is not likely possible stroke	4681
1	Patient with Possible Stroke	249

2.2. Balancing Data with PCA-KMeans

PCA-KMeans is a data preprocessing technique that combines two methods, namely Principal Component Analysis (PCA) and KMeans clustering. PCA, a statistical method, reduces the dimensionality of data by transforming a set of observations of possibly correlated variables into a set of linearly uncorrelated variable values known as principal components. [8].

KMeans clustering is a data analysis method used to group data into several clusters. KMeans clustering aims to produce clusters where each data point in a cluster has the same characteristics by calculating the distance between each data point and the centroid [16], [17]. Here, we don't directly use the KMeans function as a cluster method; instead, we use it to create clusters of PCA process results for interpretation [18].

2.3. Balancing Data with ADASYN

Adaptive Synthetic Sampling (ADASYN) is a sampling approach to the imbalanced learning problem. This method generates synthetic data samples for minority classes in the dataset to address the class imbalance problem [9]. The main idea of ADASYN is to use a weighted distribution for instances of minority classes based on the difficulty of learning [19]. It generates more synthetic data for difficult-to-learn minority class examples compared to easier-to-learn examples. ADASYN aims to reduce the bias introduced by class imbalance and shift the classification decision boundary towards difficult examples, improving learning performance [20].

2.4. Classification with Deep Neural Network

Deep Neural Networks (DNN) are machine learning algorithms inspired by the structure and function of biological neural networks in the human brain [21]. DNN is also a type of Artificial Neural Network (ANN) that consists of several layers of interconnected artificial neurons to process data and solve complex problems [22].

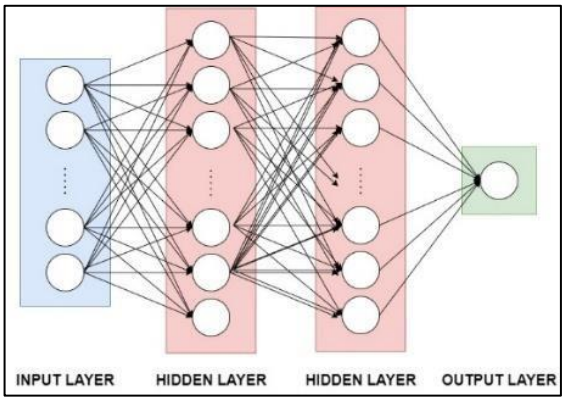

Figure 2. DNN Architecture

Figure 2 shows the architecture of a Deep Neural Network, DNN, with the first layer as the input layer [23]. This process is considered a flattening layer that converts the input into a 1- dimensional vector. In DNN, Hidden Layer 1 is the layer between the input layer and output layer. It processes the input from the previous layer and passes it on to the next layer [24]. This layer uses a dense layer type that connects neurons connected to the nearest neuron, except neurons connected to the same neuron. The number of neurons used in this layer is 128 with the activation function ReLu (Rectified Linear Unit). Hidden layer 2 is the second layer after hidden layer 1, whose content is the same as hidden layer 1, with a number of neurons of 64, which aims to gradually reduce the complexity of the model. This layer continues to use the ReLu activation function. The output layer is the last layer in the DNN, providing the final result of all data processing by the artificial neural network. The output layer can have one or more nodes [25]. The type of layer used is a dense layer, with the number of neurons adjusted to the number of classes or features in the classification task. This output layer uses a softmax activation function.

2.5. Focal Loss Function

Deep neural networks (DNNs) use the Focal loss function to handle class imbalance during training. Focal loss uses a modulated cross-entropy loss term to focus learning on difficult misclassification instances. [5].

3. RESULTS AND DISCUSSION

This point will explain how to implement and evaluate the performance of the PCA-KMeans and ADASYN data balancing techniques using DNN classification with focal loss.

3.1. Preprocessing

The dataset will be subjected to some preprocessing, namely label encoding, which aims to convert categorical data types in columns that have two labels into binary data; one hot encoding, which aims to convert categorical data types in columns that have more than two labels into binary data and new columns; outlier removal, which aims to eliminate redundancy data; checking for missing values; and the data normalization process, which aims to equalize the range of values in the dataset. The preprocessed datasets differ from the initial dataset in terms of the number of columns and data samples. Table 2 shows the preprocessing result dataset.

Table 2. Preprocessing Dataset Result

Class	Description	Initial Data	Total Data	Percentage
0	Patient is not a possible stroke	4681	3481	74.4%
1	Patient with Possible Stroke	249	208	83.5%

3.2. PCA-KMeans Data Balancing Analysis

To handle the scale differences between features, we will standardize the preprocessed data so that the feature values have a mean of zero and a standard deviation of one. Figure 3 shows the data graph before and after data standardization in the moderate imbalance case study.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

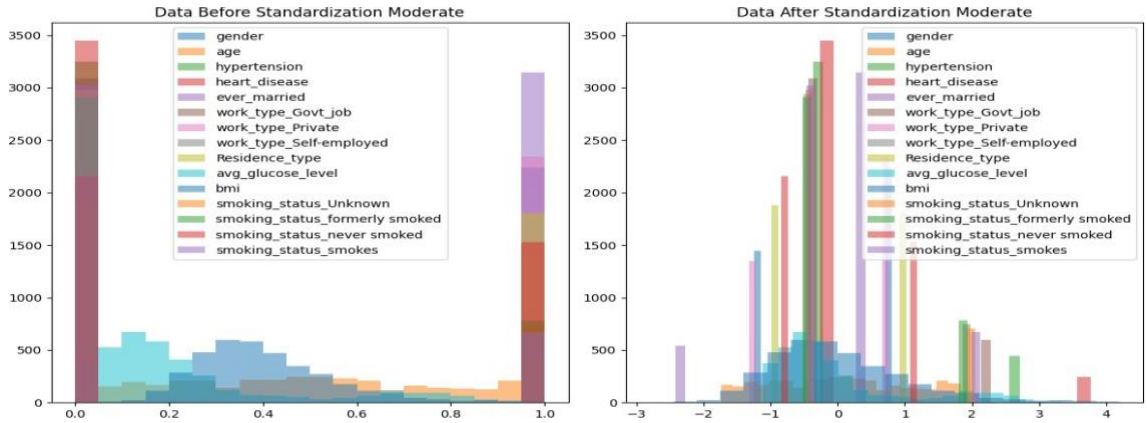

Figure 3. Data Standardization

After standardization, the graph shows a change in the range of values and a wide distribution in the data. After the data is standardized, we will proceed with the eigenvalue calculation process. Eigenvalues quantify the extent to which each main component of the data can explain variability. Table 3 displays the eigenvalues for each variable in the moderate case.

Table 3. Eigenvalues

Gender	2,0754
Age	1,6046
Hypertension	1,5222
heart_disease	1,2842
ever_married	1,1997
work_type_govt_job	1,1725
work_type_private	1,0642
work_type_self-employed	1,0087
residence_type	0,9804
avg_glucose_level	0,9241
Bmi	0,8485
smoking_status_unknown	0,7686
smoking_status_formerly smoked	0,55
smoking_status_never smoked	0
smoking_status_smokes	0

According to the table, the decreasing eigenvalues represent the largest to smallest variation in features. The gender feature has a high contribution to the total data variation, while smoking_status_never smoked and smoking_status_smokes have a value of 0, which means they have no contribution to the data variation. The selection of the number of main components using the elbow method based on eigenvalue information resulted in 13 main components that explained 100% of the data variation. Next is the PCA process, which produces the plot shown in Figure 4.

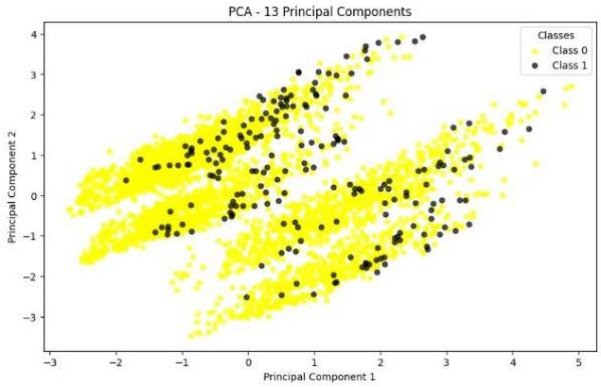

Figure 4 Plotting result of PCA

In the moderate case study, Figure 4 shows the use of 13 principal components based on the elbow method. The distribution of data generated by the PCA process is not evenly distributed but tends to overlap. To overcome this, a KMeans process is needed, which aims to find patterns or clusters that are difficult to identify by separating based on the main components. By using the elbow method and

evaluation metrics to find the optimal number of clusters and silhoutte scores, the results obtained are 6 clusters and 30 silhoutte scores. Figure 5 illustrates the KMeans plotting process.

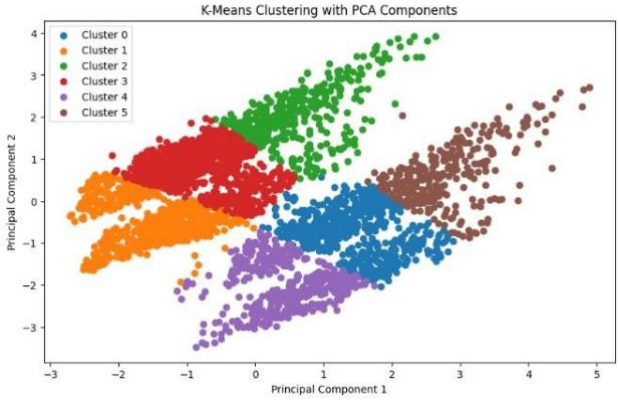

Figure 5. Plotting result of Kmeans Process

Figure 5 shows the plotting results of the KMeans process with six clusters and an optimal sihoutte value of 30. This clustering is based on the distribution patterns formed in the data, including clusters representing minority classes. This cluster identification can focus on understanding and processing data from the minority class for classification. The PCA-KMeans result data will be undersampled using near miss by reducing the imbalance between the majority class and the minority class in the clustering result data by selecting samples from the majority class that have a closer distance to the minority class samples. Figure 6 shows the plotting results using near misses.

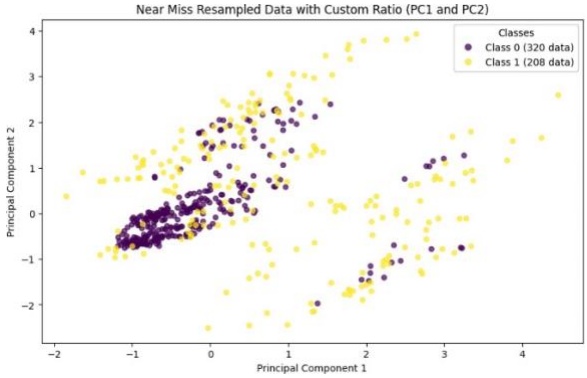

Figure 6. Plot results using Near Miss

Based on Figure 6, the majority class is not stroke (0), which previously was 3481 data to 320 data. With the decrease in the number of data samples in the majority class, namely (0), it does not cause information loss because the use of near miss still retains significant information from the majority class, which has a closer distance to the minority class sample.

3.3. ADASYN's Data Balancing Analysis

ADASYN will process the preprocessing data, assigning more weight to the minority class and generating new synthetic data examples based on the distance between the existing minority data examples. Figure 7 is a visualization of the Euclidean distance in ADASYN in the moderate case.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

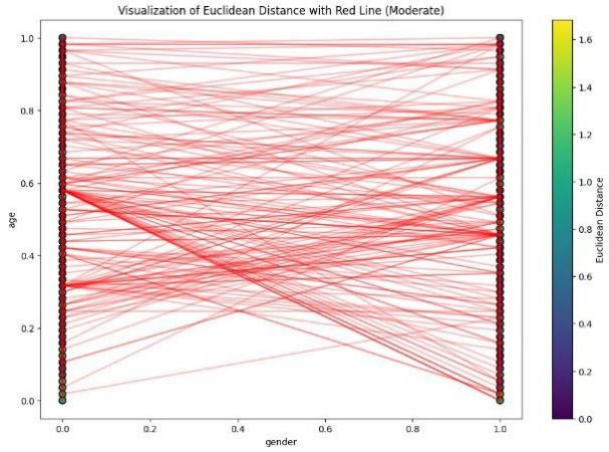

Figure 7. Visualization of Euclidean Distance Distance in ADASYN

Figure 7 is an example of Euclidean distance visualization for gender and age features. To display the points based on the two features, we use the scatter plot, with the color reflecting the Euclidean distance value to the first neighbor. The red line connects each point to its nearest neighbor to give an idea of how close or far each sample is from its neighbors. The target ADASYN ratio for the minority class is to reach a mild level of 20–40% of the dataset. The ratio between the majority and minority classes in the preprocessing data is 16.7356, which indicates that the higher the ratio value, the more unbalanced the class distribution. Therefore, the desired ratio parameter is set to target the desired ratio between the minority class and the majority class after oversampling. The process of increasing the ratio determines that the minority class represents 0.65, or 65%, of the samples from the majority class compared to the minority class. We have obtained the new synthetic data shown in Figure 8 based on the set parameters.

```
Minority Sample Size Before Oversampling (Stroke=1): 208
Number of Minority Samples After Oversampling (Stroke=1): 2290
```
Figure 8. Data Before and After ADASYN

3.4. Split of Training and Test Data

We will divide the dataset for the classification method into two parts: training data and testing data. The division of the amount of data is shown in Table 4.

Table 4. Total Data in the Model

Tier Imbalanced data	Class	Target Label	Train Data	Test Data	Total Train Data	Total Test Data	Total Data in the Model
Extreme	Model Without Data Balancing	0	2784	697	3481	30	3511
		1	24	6			
	PCA-KMeans Data Balancing Model	0	37	9	60	16	76
		1	23	7			
	ADASYN Data Balancing Model	0	2782	699	4584	1147	5731
		1	1802	448			
Moderate	Model Without Data Balancing	0	2789	692	2951	738	3689
		1	162	46			
	PCA-KMeans Data Balancing Model	0	253	67	422	106	528
		1	169	39			
	ADASYN Data Balancing Model	0	2788	693	4616	1155	5771
		1	1828	462			
Mild	Model Without Data Balancing	0	2788	693	4616	1155	5771
		1	1828	462			
	PCA-KMeans Data Balancing Model	0	2788	693	4616	1155	5771
		1	1828	462			
	ADASYN Data Balancing Model	0	2788	693	4616	1155	5771
		1	1828	462			

3.5. Parameter Setting for Focal Loss

We use Focal Loss to address the issue of imbalanced classes in classification tasks. In focal loss, there are several parameters, namely the Gamma parameter, which aims to control the extent to which focal loss penalizes misclassified samples. The higher the gamma value, the greater the penalty for

misclassification. The Alpha parameter aims to weigh the contribution of positive and negative classes. If the positive class is severely underweighted, the alpha value can be increased to give it more weight.

The gamma and alpha parameters used are 3.0 and 0.5. This is because it determines the priority for the minority class with the test results of f1-score values of 84% and 77% for classes 0 and 1, validation loss of 0.310 and 0.2048 for classes 0 and 1, and validation accuracy of 77.71% and 85.28% for classes 0 and 1. In addition, the epoch determined is 50, because after epoch 50 there is an indication of overfitting; the validation loss value increases, but the F1-Score value tends to have no increase.

3.6. Test Results on DNN Models Without Data Balancing

When testing the model using a preprocessed dataset consisting of 15 features and 1 target column, the extreme case has 2808 training data samples and 703 testing data; the moderate case has 2951 training data samples and 738 training data samples; and the mild case study has 4616 training data samples and 1155 data samples. Table 5 shows the test results without data balancing in each case.

Table 5. Model Testing Results Without Data Balancing

Imbalanced Data Tier	Class	Accuracy	Precision	Recall	F1-Score	Epoch	Train Accuracy	Valid Accuracy	Train Loss	Valid Loss
Extreme	0	96%	99%	97%	98%	1	78,67%	56,05%	0,0118	0,6591
	1		4%	17%	7%	50	96.66%	95,05%	0,0009	0,2671
Moderate	0	79%	95%	82%	88%	1	55,89%	68,7%	0,0731	0,6089
	1		13%	39%	19%	50	82,92%	80,37%	0,00334	0,4239
Mild	0	84%	94%	79%	86%	1	68,77%	73,27%	0,4662	0,5253
	1		75%	92%	82%	50	87%	84,24%	0,2136	0,3697

Table 5 shows the classification report for each case. In extreme and moderate cases, the classification report is high in class 0 and inversely proportional to class 1. In addition, the loss value in both case studies tends to be low, which slows down its decline. Without data balancing, the mild model, which has a balanced F1-score value between class 0 and class 1, emerges as the best model from the three imbalance level case studies.

3.7. Test Results on DNN Model with Focal Loss Using PCA-KMeans Data Balancing

To test the model with a dataset that uses PCA-KMeans data balancing and has 13 features, 1 cluster column, and 1 target column, there are 60 training data samples and 16 testing data samples in the extreme case studies. There are 422 training data samples and 106 training data samples in the moderate case studies, and there are 4616 training data samples and 1155 testing data samples in the mild case studies. Table 6 shows the results of the PCA-KMeans data balancing test in each case study.

Table 6. PCA-KMeans Data Balancing Model Testing Results

PCA-KMeans Data Balancing Test Results										
Imbalanced Data Tier	**Class**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Epoch**	**Train Accuracy**	**Valid Accuracy**	**Train Loss**	**Valid Loss**
Extreme	0	88%	82%	100%	90%	1	66,67%	75%	0,0307	0,0267
	1		100%	71%	83%	50	100%	100%	0,0006	0,0004
Moderate	0	89%	91%	91%	91%	1	66,47%	81,18%	0,0331	0,0297
	1		85%	85%	85%	50	97,92%	87,06%	0,0036	0,0569
Mild	0	87%	91%	86%	89%	1	72,72%	77,71%	0,0271	0,0302
	1		81%	87%	84%	50	94,37%	87,55%	0,0074	0,0336

Table 6 shows the classification report for each case. Extreme and mild cases generate classification reports that are less balanced in class 0 and 1 precision. The moderate case generates classification reports that are both balanced and consistent between class 0 and 1 precision. In addition, the loss value in the three cases tends to decrease, indicating that the model learns from errors. Using PCA-KMeans data balancing, we obtained the best model from the three cases of imbalance, which is a moderate case with a balanced F1-score value between class 0 and class 1.

3.8. Test Results on DNN Model with Focal Loss Using ADASYN Balancing

We test the model using a dataset using ADASYN data balancing, which includes 15 features and 1 target column. In the extreme case, we have 4596 training data samples and 1150 testing data, while in the moderate case, we have 4616 training data samples and 1155 training data samples. In the mild case studies, we don't conduct any testing because we use data from the moderate class. Table 7 shows the ADASYN data balancing test results for each case.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

Table 7. Table of ADASYN Data Balancing Model Testing Results

ADASYN Data Balancing Testing Results										
Imbalanced Data Tier	**Class**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Epoch**	**Train Accuracy**	**Valid Accuracy**	**Train Loss**	**Valid Loss**
Extreme	0	95%	94%	98%	96%	1	83,92%	89,13%	0,0188	0,0164
	1		96%	91%	93%	50	96,68%	93,04%	0,0043	0,0084
Moderate	0	85%	92%	83%	87%	1	70,59%	76,19%	0,0285	0,0317
	1		77%	89%	83%	50	87,78%	85,02%	0,0135	0,0251

Table 7 shows the classification report for each case. In extreme cases, it produces a balanced and optimal classification report on classes 0 and 1. In the moderate case, it generates an unbalanced classification report on classes 0 and 1 precision. Using ADASYN data balancing, the extreme model with a balanced F1-score value between class 0 and class 1 emerges as the best model in both cases of imbalance.

4. CONCLUSION

This research aims to determine the balanced dataset and optimal classification result from the Deep Neural Network model testing process with focal loss using PCA-KMeans and ADASYN data balancing techniques in the classification of possible stroke diseases. From the result, we can find the best techniques for increasing the model performance. The results are shown below.

In classification without data balancing, the F1-score difference between class 0 (not likely stroke) and class 1 (likely stroke) is very large, indicating that class 1 performs worse than class 0. As we can see in Table 5 for the extreme case, the F1-score for class 0 is 98%, and for class 1, it is 17%. In the moderate case, the F1-score is 88% for class 0 and 19% for class 1.

In classification with PCA-KMeans data balancing, the F1-score difference between class 0 and class 1 is moderate, but the precision difference is less in the moderate case than in the other cases. The loss value in the three case studies tends to decrease, indicating that the model learns from errors. The PCA-Kmeans data balancing also contributes to balancing the F1-score compared to the initial unbalanced dataset model. The average F1-score difference is only 6%, indicating a large contribution.

In classification with ADASYN data balancing, the F1-score difference between class 0 and class 1 is small, but the precision difference is less in the extreme case than in the moderate case. ADASYN data balancing also helps to balance the F1-score in comparison to the initial unbalanced dataset model. The average F1-score difference is only 3.5%, indicating a larger contribution compared to the PCA-KMeans technique.

ACKNOWLEDGEMENTS

This research is supported by the Informatics Study Program of the Institut Teknologi Nasional Bandung which has given us the opportunity to conduct this research.

REFERENCES

[1] M. Lutfi, A. T. Arsanto, M. F. Amrulloh, and U. Kulsum, "Penanganan Data Tidak Seimbang Menggunakan Hybrid Method Resampling Pada Algoritma Naive Bayes Untuk Software Defect Prediction," INFORMAL Informatics J., vol. 8, no. 2, p. 119, 2023.
[2] S. Mutmainah, "Penanganan Imbalance Data Pada Klasifikasi Kemungkinan Penyakit Stroke," J. Sains, Nalar, dan Apl. Teknol. Inf., vol. 1, no. 1, pp. 10–16, 2021.
[3] Google, "Imbalanced Data." [Online]. Available: https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data?hl=en. [Accessed: 22-Apr-2024].
[4] K. Pykes, "Oversampling and Undersampling," 2020. [Online]. Available: https://towardsdatascience.com/oversampling-and-undersampling-5e2bbaf56dcf. [Accessed: 22-Apr-2024].
[5] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 318–327, 2020.
[6] Y. Jing, "Machine Learning Performance Analysis to Predict Stroke Based on Imbalanced Medical Dataset," CAIBDA 2022 - 2nd Int. Conf. Artif. Intell. Big Data Algorithms, pp. 462–468, 2022.
[7] N. G. Ramadhan, "Comparative Analysis of ADASYN-SVM and SMOTE-SVM Methods on the Detection of Type 2 Diabetes Mellitus," Sci. J. Informatics, vol. 8, no. 2, pp. 276–282, 2021.
[8] C. Ding, "K -means Clustering via Principal Component Analysis," 2004.
[9] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," Proc. Int. Jt. Conf. Neural Networks, no. March, pp. 1322–1328, 2008.
[10] D. Yadav, "Categorical encoding using Label-Encoding and One-Hot-Encoder," 2019. [Online]. Available: https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd.

[Accessed: 22-Apr-2024].
[11] C. GOYAL, "Outlier Detection & Removal | How to Detect & Remove Outliers," 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/feature-engineering-how-to-detect-and-remove-outliers-with- python-code/. [Accessed: 22-Apr-2024].
[12] N. Sharma, "Ways to Detect and Remove the Outliers," 2018. [Online]. Available: https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba. [Accessed: 22-Apr-2024].
[13] N. Tamboli, "Effective Strategies for Handling Missing Values in Data Analysis," 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/. [Accessed: 22-Apr-2024].
[14] Google, "Normalization," 2024. [Online]. Available: https://developers.google.com/machine-learning/data-prep/transform/normalization. [Accessed: 22-Apr-2024].
[15] Kaggle, "Stroke Prediction Dataset," 2023. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset. [Accessed: 24-Apr-1BC].
[16] I. Dabbura, "K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks," 2018. [Online]. Available: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and- drawbacks-aa03e644b48a. [Accessed: 24-Apr-1BC].
[17] E. Ecosystem, "Understanding K-means Clustering in Machine Learning," 2018. [Online]. Available: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1. [Accessed: 24-Apr-1BC].
[18] E. Kaloyanova, "How to Combine PCA and K-means Clustering in Python?," 2024. [Online]. Available: https://365datascience-com.translate.goog/tutorials/python-tutorials/pca-k-means/?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc&_x_tr_hist=true. [Accessed: 24-Apr-1BC].
[19] D. V. Ramadhanti, R. Santoso, and T. Widiharih, "Perbandingan Smote Dan Adasyn Pada Data Imbalance Untuk Klasifikasi Rumah Tangga Miskin Di Kabupaten Temanggung Dengan Algoritma K-Nearest Neighbor," J. Gaussian, vol. 11, no. 4, pp. 499–505, 2023.
[20] S. Rahayu, "Analisis Perbandingan Metode Over-Sampling Adaptive (ADSYN-kNN) untuk Data dengan Fitur Nominal-Multi Categories," Citee, pp. 296–300, 2017.
[21] B. K, "Introduction to Deep Neural Networks," 2023. [Online]. Available: https://www.datacamp.com/tutorial/introduction-to-deep-neural-networks. [Accessed: 24-Apr-1BC].
[22] Binus University, "Mengenal 3 Jenis Neural Network Pada Deep Learning," 22AD. [Online]. Available: https://sis.binus.ac.id/2022/04/21/mengenal-3-jenis-neural-network-pada-deep-learning. [Accessed: 24-Apr-2024].
[23] M. Rouse, "Input Layer," 2018. [Online]. Available: https://www.techopedia.com/definition/33262/input-layer-neural-networks. [Accessed: 22-Apr-2024].
[24] P. Antoniadis, "Hidden Layers in a Neural Network," 2024. [Online]. Available: https://www.baeldung.com/cs/hidden-layers-neural-network. [Accessed: 24-Apr-2024].
[25] R. Fajri, "Neural Network: Algoritma yang Menjadi Inti dari ChatGPT," 2023. [Online]. Available: https://www.dicoding.com/blog/neural-network-algoritma-yang-menjadi-inti-dari-chatgpt. [Accessed: 24-Apr-2024].