SINTECHCOM: Science, Technology and Communication Journal
Vol. 7, No. 1, October 2026, pp. 53-68, DOI: 10.59190/stc.v7i1.438

The impact of ADASYN on Alzheimer's disease classification performance using backpropagation neural networks

Alghozi Rafif Ali, Iis Afrianty*, Pizaini Pizaini, Yusra Yusra, Nazruddin Safaat Harahap
Department of Informatics Engineering, UIN Sultan Syarif Kasim Riau, Pekanbaru 28293, Indonesia

ABSTRACT

Alzheimer's disease (AD) is a neurodegenerative disorder that causes a decline in cognitive ability and the capacity to perform daily activities. Class imbalance in Alzheimer's data may bias machine learning models toward the majority class and reduce their ability to recognize the minority class. This study applied adaptive synthetic sampling (ADASYN) to a backpropagation neural network (BPNN) to balance the class distribution and improve the classification of Alzheimer's disease. The Kaggle dataset contained 2,149 records (760 Alzheimer and 1,389 non-Alzheimer) with 34 input variables, which were reduced to 32 after selection, and one binary output variable. After preprocessing, the data were split using 10-fold cross-validation. Z-score normalization and ADASYN were applied to the training data of each fold, whereas the test data received only Z-score normalization. The 32-33-1, 32-48-1, and 32-63-1 architectures were evaluated with learning rates of 0.01, 0.05, and 0.09, each with and without ADASYN. The 10-fold average results showed that ADASYN consistently improve neither accuracy (80.47% with ADASYN versus 80.41% without) nor recall, which was higher in only 2 of 9 configurations (70.83% versus 72.28%). The best configuration was the 32-63-1 architecture with a learning rate of 0.01 and ADASYN, which achieved 81.57% accuracy, 75.05% precision, 71.84% recall, and 73.31% F1-score. ADASYN tended to reduce false positives but did not reduce undetected Alzheimer cases (false negatives) providing no better results for Alzheimer's disease diagnosis.

ARTICLE INFO

Article history:

Received Sep 11, 2026
Revised Oct 16, 2026
Accepted Oct 23, 2026
Published Oct 31, 2026

Keywords:

ADASYN
Alzheimer's Disease
BPNN
Classification
Machine Learning

Copyright (c) 2026:

Alghozi Rafif Ali, Iis Afrianty, Pizaini Pizaini, Yusra Yusra, Nazruddin Safaat Harahap

License:

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY).

*Corresponding Author
E-mail address: iis.afrianty@uin-suska.ac.id

1. INTRODUCTION

Alzheimer's disease is a progressive neurodegenerative disease characterized by cognitive decline and a diminishing ability to perform daily activities [1]. In 2021, approximately 49 million adults aged ≥ 65 years lived with Alzheimer's disease and related dementias, a 160% increase from 18.7 million people in 1991 [2]. Symptoms of Alzheimer's disease, such as the loss of basic skills and abilities, make the early diagnosis of patients difficult and inaccurate [3]. Therefore, early detection of the disease cannot be overlooked, as it highly contributes to prompt management [4].

Alzheimer's research has widely employed machine learning and deep learning to improve classification accuracy [5]. Detection of Alzheimer's disease network (DAD-Net), applied to medical images, was reported to achieve an accuracy of 99.22%, precision of 99.30%, and recall of 99.14%

[6], whereas the application of backpropagation neural network (BPNN) to ^{18}F-florbetapir PET images from the ADNI database yielded an accuracy of 87.9% for the classification of normal and Alzheimer's (NC/AD) groups [7]. A back-propagation neural network trained on clinical and biomarker variables achieved an accuracy of 92.13% for Alzheimer's disease diagnosis, outperforming logistic regression, k-nearest neighbors, and support vector machine [8]. These differences show that performance is strongly influenced by the characteristics of the data, algorithm, and evaluation scheme; therefore, the BPNN is used because it can learn nonlinear relationships among features through a weight-updating process. A study using a BPNN demonstrated superior performance, achieving a maximum accuracy of 96.14% in stroke classification [9]. The accuracy in an earlier study was relatively high compared with a diabetes classification study, which obtained an accuracy of 83% [10]. However, BPNN remains biased toward the majority class when the training data are imbalanced [11]. Previous studies have not specifically compared BPNN with and without data balancing for Alzheimer's disease classification under identical testing configurations.

Adaptive synthetic sampling (ADASYN) is an oversampling technique that generates samples for the minority class, providing more samples to areas that are harder to learn [12]. Previous study on migraine classification reported that ADASYN improved the accuracy of a stochastic gradient boosting model from 86.87% to 94.82% on an imbalanced seven-class dataset [13]. The combination of ADASYN and BPNN has been applied to stroke classification, whereas without ADASYN, the model achieved a high accuracy of 94.52% but a recall of only 4%, indicating that it failed to identify most stroke cases, whereas the recall increased to 93.45% after ADASYN was applied [14]. The implementation of ADASYN together with DAD-Net in Alzheimer's disease classification showed high performance [6]. In lung cancer prediction, the combination of ADASYN and Support Vector Machine (SVM) produced an accuracy of 98.95% [15], while a comparison of SMOTE and ADASYN in classifying poor households showed that ADASYN can be used to improve learning on imbalanced data [16].

This study applied ADASYN to BPNN and compared its performance with that of a conventional BPNN to determine the effect of implementing data balancing on Alzheimer's disease classification.

2. RESEARCH METHODS

This study was conducted in accordance with the research stages shown in Figure 1. The process begins with data collection initialization, in which an Alzheimer's disease dataset of 2,149 records was obtained from Kaggle. Data selection was then performed to remove irrelevant attributes and separate the target variables. The selected data were then preprocessed in the form of missing values and outlier checks. The processed data were stratified and split using K-Fold Cross-Validation. In each fold, the data were transformed using Z-score normalization computed from the training data and applied to the fold test data. Subsequently, ADASYN was deployed on each fold of the training data to resolve data imbalance before training and evaluation. Balanced data were used to train the BPNN model, which was then evaluated using a confusion matrix on the unbalanced fold test data. Each scenario was run under two conditions: with and without ADASYN to compare the effect of data balancing.

2.1. Data Collection Initialization

The research dataset are secondary Alzheimer's disease data obtained from Kaggle at https://www.kaggle.com/datasets/rabieelkharoua/alzheimers-disease-dataset/data. The dataset contained 2,149 records with 35 columns, consisting of 34 input variables and a target variable, Diagnosis. The class distribution was imbalanced, consisting of 760 Alzheimer's and 1,389 non-Alzheimer's records.

The 34 input variables are PatientID, Age, Gender, Ethnicity, EducationLevel, BMI, Smoking, AlcoholConsumption, PhysicalActivity, DietQuality, SleepQuality, FamilyHistoryAlzheimers, CardiovascularDisease, Diabetes, Depression, HeadInjury, Hypertension, SystolicBP, DiastolicBP, CholesterolTotal, CholesterolLDL, CholesterolHDL, CholesterolTriglycerides, MMSE, FunctionalAssessment, MemoryComplaints, BehavioralProblems, ADL, Confusion, Disorientation, PersonalityChanges, DifficultyCompletingTasks, Forgetfulness, and DoctorInCharge. An example of the raw data is presented in Table 1.

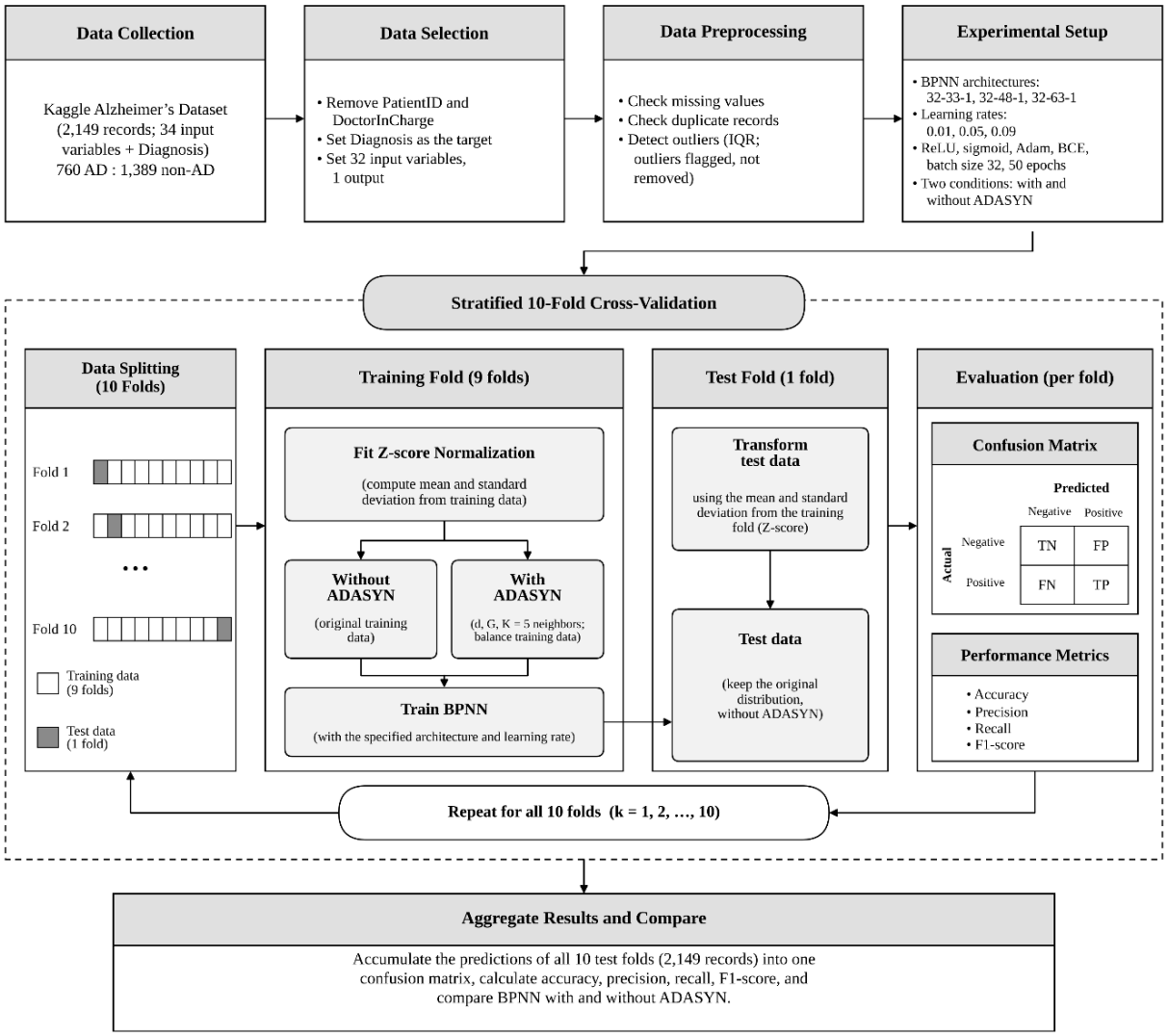


Figure 1. Research methodology.

Table 1. Raw data.

PatientID	Age	Gender	⋯	Forgetfulness	Diagnosis	DoctorInCharge
4751	73	0	⋯	0	0	XXConfid
4752	89	0	⋯	1	0	XXConfid
…	…	…	…	…	…	…
6898	78	1	⋯	1	1	XXConfid
6899	72	0	⋯	1	0	XXConfid

2.2. Data Selection

Data selection was performed by removing identity and administrative attributes that did not contribute to classification, namely PatientID and DoctorInCharge, and then separating the Diagnosis variable as the target. Consequently, the data, which originally had 34 variables and 1 output, became 32 input variables and 1 output, with class 1 for Alzheimer and class 0 for Non-Alzheimer. The selected data were then preprocessed.

2.3. Data Preprocessing

Data preprocessing was performed to ensure data quality before the data were split. At this stage, all variables were checked for missing values to identify empty or unrecorded entries in the dataset. If missing values are found, the handling method is determined by the variable type, for exam-

ple, median imputation for numerical data or mode imputation for categorical data. In addition, duplicate records were checked, and outliers in continuous variables were detected using the Interquartile Range (IQR), as shown in Equation (1) and Equation (2).

$$IQR = Q_3 - Q_1 \tag{1}$$

$$B_{lower} = Q_1 - 1.5 \times IQR, \quad B_{upper} = Q_3 + 1.5 \times IQR \tag{2}$$

Q_1 and Q_3 are the first and third quartiles, respectively. A value is flagged as an outlier if it lies below the lower or above the upper bound. Extreme values were not removed automatically because the data were removed only when proven invalid. Extreme values that remain clinically plausible may provide important information [17].

2.4. K-Fold Data Splitting

The pre-processed data (2,149 records) were split using stratified K-fold cross-validation with K = 10 folds, so that the proportions of the Alzheimer and non-Alzheimer classes in each fold were preserved. The data were divided into 10 parts with 10 iterations. In each iteration (fold), one part is used as the test data and the remaining nine parts as the training data, such that each record serves as the test data exactly once. The training data of each fold were transformed and balanced with ADASYN, whereas the fold test data were only transformed and not balanced. The test data from all the folds were used for confusion matrix evaluation. The data-splitting scheme is illustrated in Figure 2.

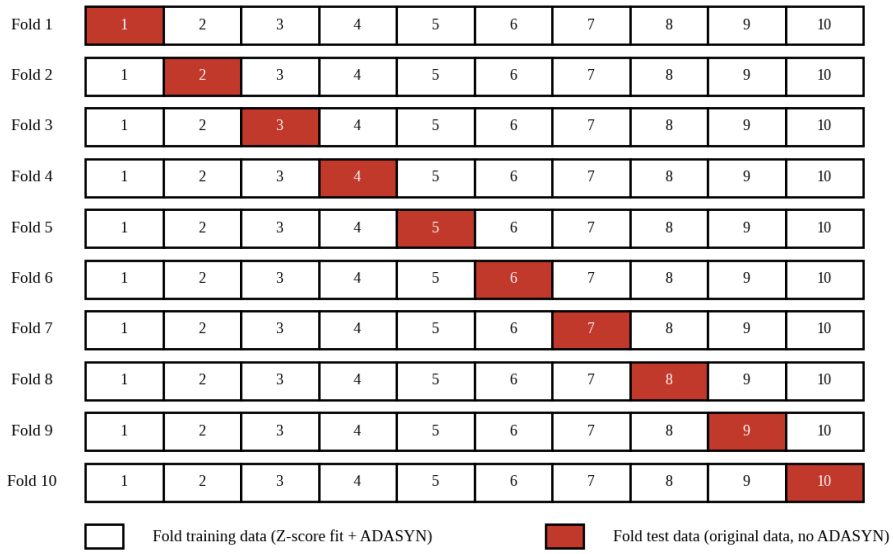


Figure 2. 10-fold cross-validation data splitting scheme.

2.5. Data Transformation

In each fold, the training and fold test data were transformed using Z-score normalization computed only from the fold training data using Equation (3) and Equation (4), after which the new values were calculated using Equation (5).

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \tag{3}$$

$$\sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \mu_j\right)^2} \tag{4}$$

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{5}$$

where x_{ij} is the value of the i-th record for the j-th variable, n is the number of fold training records, and z_{ij} is the normalized value. The same mean (μ_j) and standard deviation (σ_j) values were used to transform the fold test data so that all data were on the same scale.

2.6. ADASYN

The transformed training data were balanced using ADASYN. ADASYN adaptively generates synthetic samples for the minority class by assigning more samples to minority records that are difficult to learn to those surrounded by more majority-class members [16]. Balancing was applied only to the training data, whereas the fold test data remained the original data to prevent data leakage and guarantee an unbiased model evaluation [18]. The ADASYN process begins by calculating the degree of class imbalance (d) using Equation (6).

$$d = \frac{m_s}{m_l} \tag{6}$$

where m_s and m_l are the numbers of minority-class and majority-class records, respectively, in the training data. If d is less than the imbalance threshold, the total number of synthetic data points to be generated (G) is calculated using Equation (7).

$$G = (m_l - m_s) \times \beta \tag{7}$$

where β lies in the range of 0 to 1 and indicates the desired level of balance; β = 1 means that the two classes are fully balanced. Next, for each minority record x_i, the K nearest neighbors (K = 5) are found using the Euclidean distance in Equation (8), where P = 32 is the number of variables.

$$dist(x_i, x_k) = \sqrt{\sum_{p=1}^{P}(x_{ip} - x_{kp})^2} \tag{8}$$

The difficulty level of each minority record (r_i) is calculated with Equation (9), where Δ_i is the number of neighbors belonging to the majority class among the K nearest neighbors of x_i. The value r_i is then normalized using Equation (10) such that the sum of all $\hat{r}_i$ equals 1.

$$r_i = \frac{\Delta_i}{K} \tag{9}$$

$$\hat{r}_i = \frac{r_i}{\sum_{i=1}^{m_s} r_i} \tag{10}$$

The number of synthetic records generated around each minority record x_i is determined using Equation (11). Minority records with a large $\hat{r}_i$ receive more synthetic records.

$$g_i = \hat{r}_i \times G \tag{11}$$

For each synthetic record, one minority neighbor x_{zi} is randomly selected from the K nearest neighbors of x_i, and the synthetic record s_i is then generated by interpolation using Equation (12), where λ is a random number in the range of 0 to 1. This step is repeated until g_i synthetic records have been generated for each x_i, after which the synthetic records are combined with the original training data so that the class distribution becomes more balanced.

$$s_i = x_i + (x_{zi} - x_i) \times \lambda \tag{12}$$

2.7. Backpropagation Neural Network (BPNN)

Balanced training data were used to train the BPNN classification model. The BPNN architecture consisted of 32 input neurons corresponding to the number of variables, three alternative numbers of hidden layer neurons (33, 48, and 63), and one output neuron. The BPNN learns the relationships

among features through three phases: forward propagation, backpropagation, and weight updating [9]. Before training the data, hyperparameter optimization was performed with combinations of the learning rate and number of hidden-layer neurons, as shown in Table 2, to maximize the accuracy of the classification model.

Table 2. Hyperparameter optimization design.

BPNN architecture	Learning rate
32-33-1	α = 0.01
	α = 0.05
	α = 0.09
32-48-1	α = 0.01
	α = 0.05
	α = 0.09
32-63-1	α = 0.01
	α = 0.05
	α = 0.09

The training and testing processes of the BPNN are presented in Figure 3. During training, forward propagation computes the weighted combination and activation function, backpropagation computes the error, and the weights are updated until the stopping criterion is met. During testing, the weights obtained from training were used to generate predictions on the test data without updating the weights.

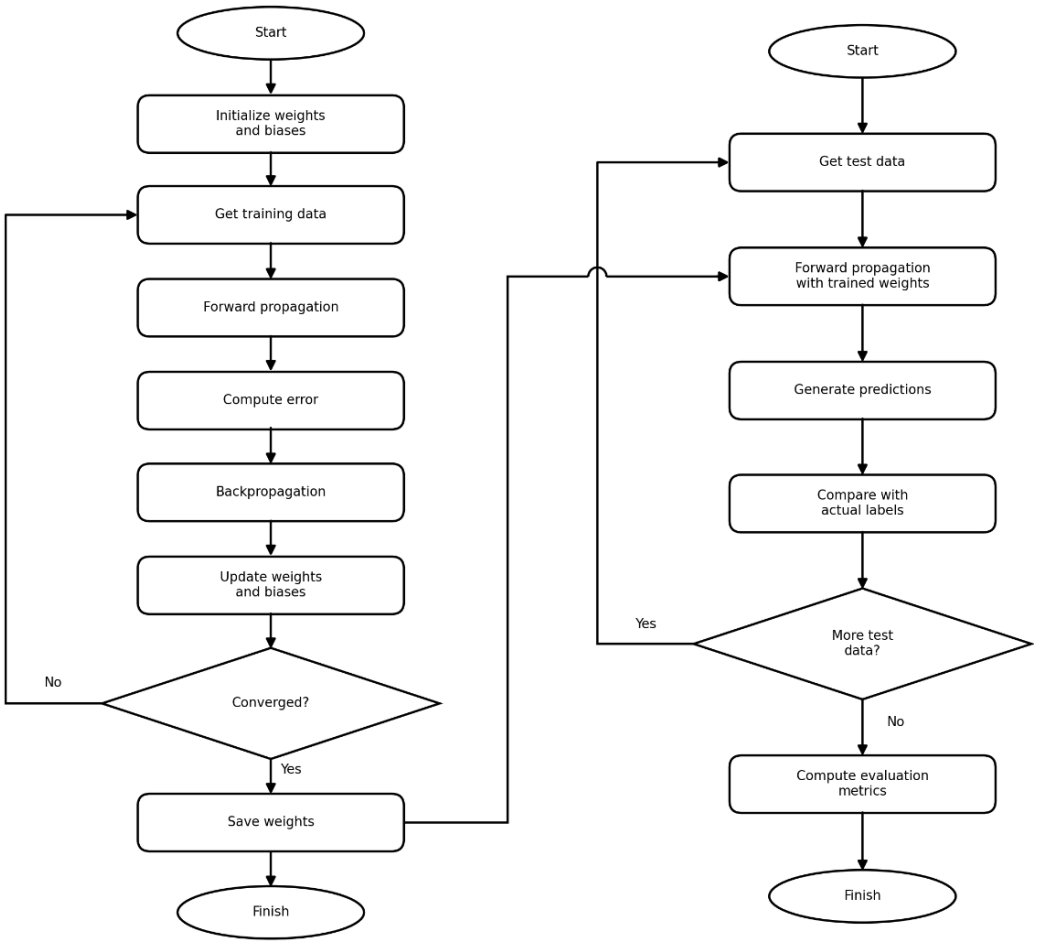


Figure 3. Flowchart of BPNN training and testing.

Forward propagation is the phase in which each neuron in the hidden and output layers computes the weighted sum of its inputs using Equation (13) and then applies the ReLU activation function in Equation (14). The output neuron computes the weighted sum of the hidden layer outputs using Equation (15) and applies the sigmoid activation function in Equation (16), producing the probability of the Alzheimer class [10].

$$z_j = \sum_{i=1}^{32} w_{ij}\, x_i + b_j \tag{13}$$

$$h_j = f(z_j) = \max(0, z_j) \tag{14}$$

$$z_k = \sum_{j=1}^{H} w_{jk}\, h_j + b_k \tag{15}$$

$$y_k = \frac{1}{1+e^{-z_k}} \tag{16}$$

where w_{ij} is the weight from the i-th input to the j-th hidden neuron, b_j is the bias, H is the number of hidden-layer neurons (33, 48, or 63), and y_k is the model output. A record is classified as Alzheimer if $y_k > 0.5$. The prediction error is measured using the binary cross-entropy loss function in Equation (17), where t_n is the actual label and N is the number of records.

$$L = -\frac{1}{N}\sum_{n=1}^{N}[t_n \log(y_n) + (1 - t_n)\log(1 - y_n)] \tag{17}$$

Forward propagation is followed by the backpropagation phase, in which the error at the output layer (δ_k) is computed using Equation (18) and propagated to the hidden layer (δ_j) using Equation (19) to determine each neuron's contribution to the prediction error. $f'(z_j)$ is the derivative of ReLU, which equals 1 if $z_j > 0$ and 0 otherwise.

$$\delta_k = y_k - t_k \tag{18}$$

$$\delta_j = f'(z_j) \sum_k \delta_k\, w_{jk} \tag{19}$$

In the final phase, weight updating, the weights and biases are updated to minimize the error based on the learning rate (α) and error gradient using Equation (20) through Equation (21) and Equation (22) [19]. In the implementation, the weights were updated using the Adam optimizer, which adjusted the update step size of each weight based on the previous gradients. The three phases were repeated for 50 epochs [20]. After training is complete, the trained weights are used to predict the test data with Equation (13) through Equation (16) without updating the weights.

$$\Delta w_{ij} = -\alpha\, \delta_j\, x_i \tag{20}$$

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij} \tag{21}$$

$$b_j^{new} = b_j^{old} - \alpha\, \delta_j \tag{22}$$

After the training process was complete, the model was tested, as shown in Figure 3, using Equation (13) and Equation (14) on the test data of each fold to determine whether the model converged.

2.8. Evaluation

The evaluation uses a confusion matrix, Table 3 that compares the model's predictions with the original labels (ground truth) of the test data, and is used to measure the performance of classification models in machine learning [21]. True Positive (TP) is Alzheimer data correctly predicted as Alzheimer, True Negative (TN) is Non-Alzheimer data correctly predicted as such, False Positive (FP) is

Non-Alzheimer data incorrectly predicted as Alzheimer (Type I error), and False Negative (FN) is Alzheimer data incorrectly predicted as Non-Alzheimer (Type II error) [22].

Table 3. Confusion matrix.

Actual\predicted	Has Alzheimer's disease	No Alzheimer's disease
Has Alzheimer's disease	True positive	False negative
No Alzheimer's disease	False positive	True negative

From the confusion matrix, the values of accuracy, precision, recall and F1 score were obtained [23]. Accuracy indicates how often the model predicts correctly, for both the positive and negative classes, as shown in Equation (23).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{23}$$

Precision measures the proportion of all records predicted as positive by the model and that are actually positive, as shown in Equation (24).

$$Precision = \frac{TP}{TP+FP} \tag{24}$$

Recall, or sensitivity, measures the proportion of positive records successfully detected by the model among all actual positive records, that is, it minimizes false-negative errors, as shown in Equation (25).

$$Recall = \frac{TP}{TP+FN} \tag{25}$$

F1-Score is the harmonic mean of precision and recall, providing a balance between the two metrics and proving useful when the class distribution is imbalanced, as shown in Equation (26).

$$F1\text{-}score = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{26}$$

The evaluation results were used to compare the models with and without ADASYN under the same architecture and learning rate configurations.

3. RESULTS AND DISCUSSIONS

This section presents the results of each research stage, from data collection, data selection, data preprocessing (missing-value and outlier checks), K-Fold data splitting, data transformation, and class balancing using ADASYN to BPNN implementation, and model performance comparison. The discussion focuses on the effect of ADASYN on accuracy and other metrics across the tested combinations of architectures and learning rates.

3.1. Data Selection

The PatientID and DoctorInCharge variables were removed because they served as identifiers and single-valued administrative attributes, whereas Diagnosis was separated as the output target. The selection resulted in 2,149 records with 32 input variables and one output. An example of the data after selection is presented in Table 4.

3.2. Data Preprocessing

In the preprocessing stage, the check of all columns revealed no missing values or duplicate records; therefore, no imputation was required, and the number of records remained 2,149. Outlier detection with the IQR was performed on 15 continuous variables, whereas the other 17 variables are

binary or categorical. As shown in Table 5, no values fell outside the lower or upper bounds; therefore, all data were retained.

Table 4. Data after selection.

Age	Gender	⋯	Forgetfulness	Diagnosis
73	0	⋯	0	0
89	0	⋯	1	0
…	…	…	…	…
78	1	⋯	1	1
72	0	⋯	1	0

Table 5. Outlier check results for continuous variables.

Variable	Q1	Q3	Lower bound	Upper bound	Outlier
Age	67.00	83.00	43.00	107.00	0
BMI	21.61	33.87	3.22	52.26	0
AlcoholConsumption	5.14	15.16	-9.89	30.19	0
PhysicalActivity	2.57	7.43	-4.72	14.71	0
DietQuality	2.46	7.56	-5.19	15.21	0
SleepQuality	5.48	8.56	0.86	13.18	0
SystolicBP	112.00	157.00	44.50	224.50	0
DiastolicBP	74.00	105.00	27.50	151.50	0
CholesterolTotal	190.25	262.03	82.58	369.70	0
CholesterolLDL	87.20	161.73	-24.61	273.54	0
CholesterolHDL	39.10	78.94	-20.67	138.70	0
CholesterolTriglycerides	137.58	314.84	-128.30	580.72	0
MMSE	7.17	22.16	-15.32	44.65	0
FunctionalAssessment	2.57	7.55	-4.90	15.02	0
ADL	2.34	7.58	-5.52	15.44	0

3.3. K-Fold Data Splitting

All 2,149 preprocessed records (760 Alzheimer and 1,389 non-Alzheimer) were split into 10 stratified folds. In folds 1 to 9, the training data comprised 1,934 records (684 Alzheimer and 1,250 non-Alzheimer) and the test data comprised 215 records (76 Alzheimer and 139 non-Alzheimer), whereas in fold 10, the training data comprised 1,935 records (684 Alzheimer and 1,251 non-Alzheimer) and the test data comprised 214 records (76 Alzheimer and 138 Non-Alzheimer). The class proportions in each fold were preserved, and each record served as test data exactly once; thus, the total test data across the 10 folds amounted to 2,149 records, as shown in Table 6.

Table 6. Data splitting results.

Data portion	Total	Alzheimer	Non-Alzheimer
All data (10-fold CV)	2,149	760	1,389
Training data per fold (9/10)	1,934 – 1,935	684	1,250 – 1,251
Test data per fold (1/10)	214–215	76	138 – 139

3.4. Data Transformation

Z-score normalization was computed from the training data of each fold. For example, in fold 1, the Age variable has μ = 74.8935 and σ = 8.9923, so the patient with PatientID 4751, Age 73 years, has a normalized value of $z = (73 - 74.8935) / 8.9923$ = -0.2106, consistent with Table 7. The MMSE variable had μ = 14.7338 and σ = 8.6000. After transformation, each variable in the fold training data had a mean of 0 and a standard deviation of 1. The same μ and σ parameters were then used to transform the corresponding fold test data. An example of the normalization results is presented in Table 7.

Table 7. Z-score normalization results on fold 1 training data.

PatientID	Age	Gender	BMI	SystolicBP	⋯	MMSE	FunctionalAssessment	ADL	Forgetfulness
4751	-0.2106	-1.0125	-0.6520	0.3243	⋯	0.7825	0.4870	-1.1054	-0.6601
4752	1.5687	-1.0125	-0.1112	-0.7164	⋯	0.6837	0.6928	-0.8115	1.5148
4753	-0.2106	-1.0125	-1.3636	-1.3332	⋯	-0.8579	0.2730	0.7239	-0.6601
4754	-0.0994	0.9877	0.8557	-0.6008	⋯	-0.0864	1.3263	0.5074	-0.6601
4755	1.5687	-1.0125	-0.9586	-1.5259	⋯	-0.1414	0.3244	-1.6858	-0.6601

3.5. ADASYN

In the fold 1 training data, there are 684 Alzheimer records and 1,250 non-Alzheimer records, so the degree of imbalance is d = 684/1,250 = 0.5472. With β = 1, the targeted number of synthetic records is G = (1,250 − 684) × 1 = 566. Because the g_i values are rounded for each minority record, ADASYN generated 594 synthetic records, so the fold 1 training data distribution became 1,278 Alzheimer records and 1,250 non-Alzheimer records, a total of 2,528 records. In each cross-validation fold, ADASYN was reapplied only to the training data of that fold, and the number of Alzheimer records after ADASYN ranged from 1,254 to 1,279, with an average total of 2,522.3 training records. The changes in the distribution are shown in Figure 4. The fold test data remained in the original 215 records (214 records in fold 10).

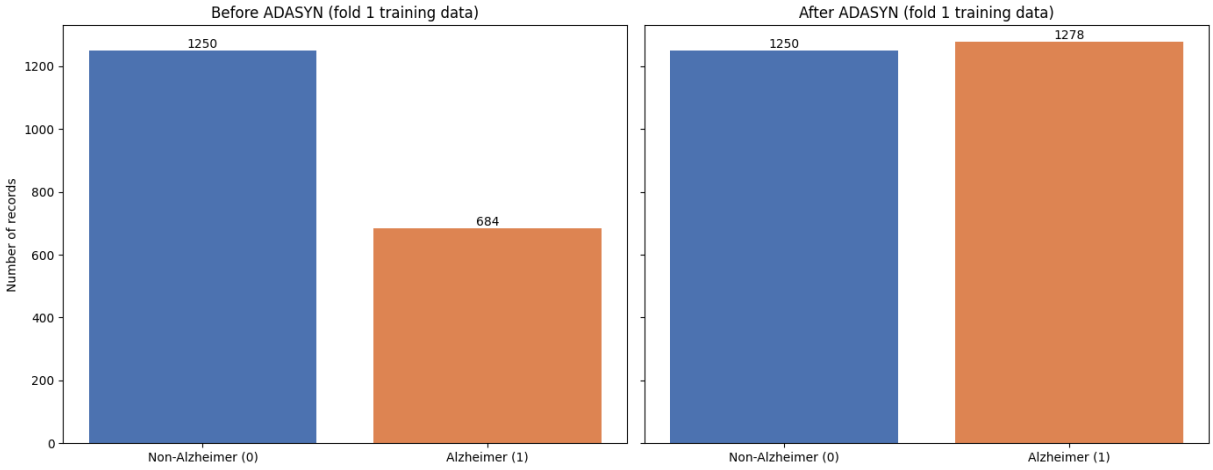


Figure 4. Class distribution of fold 1 training data before and after ADASYN.

3.6. Backpropagation Neural Network (BPNN)

The model uses the TensorFlow library through the Keras module in Python, which is implemented on Google Colab. Testing was conducted with 18 scenarios based on combinations of three architectures (32-33-1, 32-48-1, and 32-63-1), three learning rates (0.01, 0.05, and 0.09), and two data conditions (with and without ADASYN), as shown in Table 8. Every model used ReLU activation in the hidden layer, sigmoid in the output layer, the Adam optimizer, binary cross-entropy (BCE) loss, a batch size of 32, and 50 epochs.

3.7. Discussion

Table 9 presents the results of 18 testing scenarios. The accuracy values are the averages of the 10 folds computed on the test data of each fold and evaluated using a confusion matrix.

Based on Table 9, applying ADASYN to the training data did not yield consistent improvement in accuracy. On the 10-fold test data, ADASYN produced higher accuracy in four of nine configurations, namely the 32-33-1 architecture with a learning rate of 0.01, the 32-48-1 architecture with a learning rate of 0.05, and the 32-63-1 architecture with learning rates of 0.01 and 0.05. In the other five configurations, the accuracy with ADASYN was lower, with the largest difference of 1.07 per-

centage points for the 32-33-1 architecture with a learning rate of 0.09. The mean accuracy with ADASYN was 80.47%, which was nearly identical to 80.41% without ADASYN.

Table 8. Testing scenarios.

No	Architecture	Learning rate	ADASYN	K-fold	Epoch
1	32-33-1	0.01	Yes	10	50
2	32-33-1	0.01	No	10	50
3	32-33-1	0.05	Yes	10	50
4	32-33-1	0.05	No	10	50
5	32-33-1	0.09	Yes	10	50
6	32-33-1	0.09	No	10	50
7	32-48-1	0.01	Yes	10	50
8	32-48-1	0.01	No	10	50
9	32-48-1	0.05	Yes	10	50
10	32-48-1	0.05	No	10	50
11	32-48-1	0.09	Yes	10	50
12	32-48-1	0.09	No	10	50
13	32-63-1	0.01	Yes	10	50
14	32-63-1	0.01	No	10	50
15	32-63-1	0.05	Yes	10	50
16	32-63-1	0.05	No	10	50
17	32-63-1	0.09	Yes	10	50
18	32-63-1	0.09	No	10	50

Table 9. Mean accuracy test results.

No	BPNN architecture	Learning rate	Mean accuracy with ADASYN	Mean accuracy without ADASYN
1	32-33-1	0.01	81.11%	80.69%
2		0.05	80.04%	80.50%
3		0.09	78.92%	79.99%
4	32-48-1	0.01	79.94%	80.32%
5		0.05	81.34%	80.36%
6		0.09	79.43%	80.13%
7	32-63-1	0.01	81.57%	80.50%
8		0.05	81.43%	80.55%
9		0.09	80.41%	80.69%

The effect of ADASYN can be seen more clearly in recall, that is, the model's ability to recognize patients with Alzheimer's disease. Table 10 compares the recall, F1-score, and number of False Negatives (FN) in the accumulated confusion matrix of the 10 folds (760 Alzheimer records) for each pair of configurations with and without ADASYN.

Based on Table 10, ADASYN produced a higher recall in only two of nine configurations, namely the 32-48-1 architecture with a learning rate of 0.09 and the 32-63-1 architecture with a learning rate of 0.01, with a mean of 70.83% compared with 72.28% without ADASYN. The mean FN increased from 210.67 to 221.67 cases per configuration, whereas the mean FP decreased from 210.22 to 198.11, and the mean precision increased from 72.62% to 73.48%. The mean F1-score of the two conditions was relatively similar, at 71.87% with ADASYN and 72.26% without ADASYN, because the decrease in recall was only partially offset by the increase in precision. This pattern indicates that, in this dataset, the synthetic data generated by ADASYN did not shift the model's predictions toward the Alzheimer class; instead, the model tended to be more conservative, so that false alarms decreased, but missed Alzheimer cases did not. This is presumably because some synthetic data were generated in regions that overlapped with the majority class, thereby adding noise to the learning process.

Table 10. Comparison of recall, F1-score, and false negatives on testing results.

No	BPNN Architecture	Learning Rate	Recall With ADASYN	Recall Without ADASYN	F1-score With ADASYN	F1-score Without ADASYN	FN With ADASYN	FN Without ADASYN
1		0.01	71.58%	74.74%	72.63%	73.31%	216	192
2	32-33-1	0.05	70.00%	73.16%	71.09%	72.61%	228	204
3		0.09	70.00%	71.31%	70.02%	71.57%	228	218
4		0.01	70.92%	71.32%	71.39%	71.88%	221	218
5	32-48-1	0.05	70.00%	74.34%	72.58%	72.78%	228	195
6		0.09	71.05%	69.61%	70.88%	71.18%	220	231
7		0.01	71.84%	71.32%	73.31%	72.12%	214	218
8	32-63-1	0.05	71.32%	72.50%	73.01%	72.47%	218	209
9		0.09	70.79%	72.24%	71.93%	72.47%	222	211
Mean			70.83%	72.28%	71.87%	72.26%	221.67	210.67

The highest accuracy was obtained by the 32-63-1 architecture with a learning rate of 0.01 using ADASYN, at 81.57%, with a recall of 71.84% and 214 FN. Figure 5 shows the accumulated 10-fold confusion matrix of the model on 2,149 test records (all of the original data, each serving as test data exactly once), with 1,207 TN, 182 FP, 214 FN, and 546 TP. Based on Equations (23) – (26), the model obtained a mean accuracy of 81.57%, precision of 75.05%, recall of 71.84%, and F1-score of 73.31%, as shown in Figure 6. Of the 760 Alzheimer's records, 546 were correctly recognized.

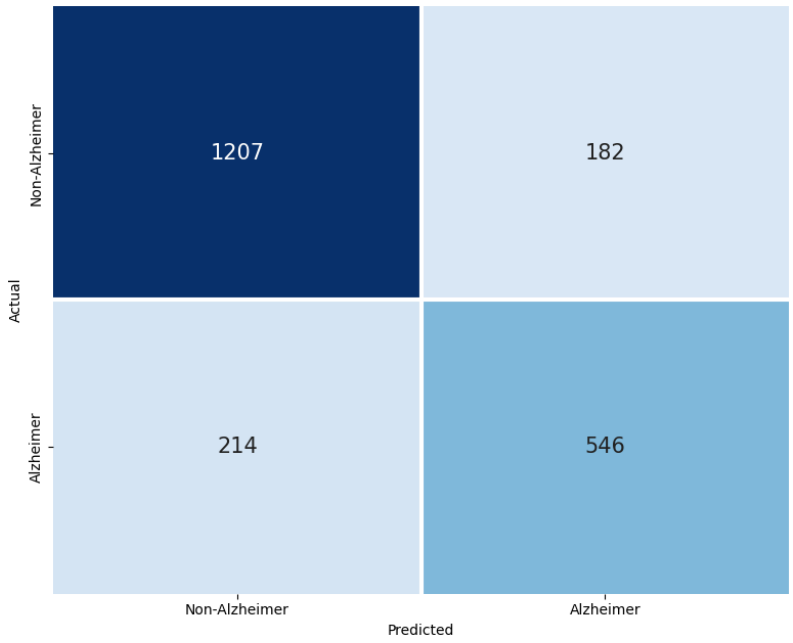


Figure 5. Accumulated confusion matrix of the 32-63-1 architecture with learning rate of 0.01 and ADASYN.

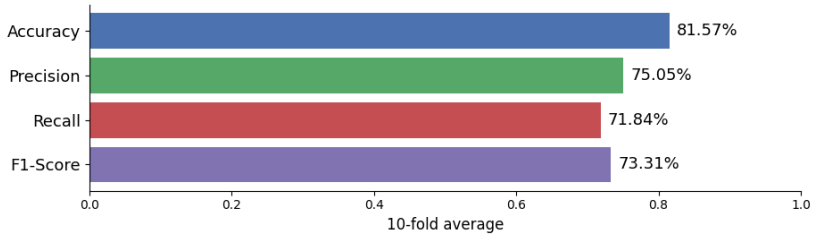


Figure 6. Mean Metrics of 32-63-1 architecture with learning rate of 0.01 and ADASYN.

For comparison, Figure 7 shows the accumulated confusion matrix of the same configuration without ADASYN, with 1,188 TN, 201 FP, 218 FN, and 542 TP. The model produced a mean accuracy of 80.50%, precision of 72.95%, recall of 71.32%, and F1-score of 72.12%, as shown in Figure 8. Accordingly, in the same configuration, ADASYN increased the accuracy by 1.07 percentage points and recall by 0.52 percentage points, and reduced the number of FN from 218 to 214 and FP from 201 to 182. The lowest accuracy of 78.92% was obtained by the 32-33-1 architecture with a learning rate of 0.09 using ADASYN.

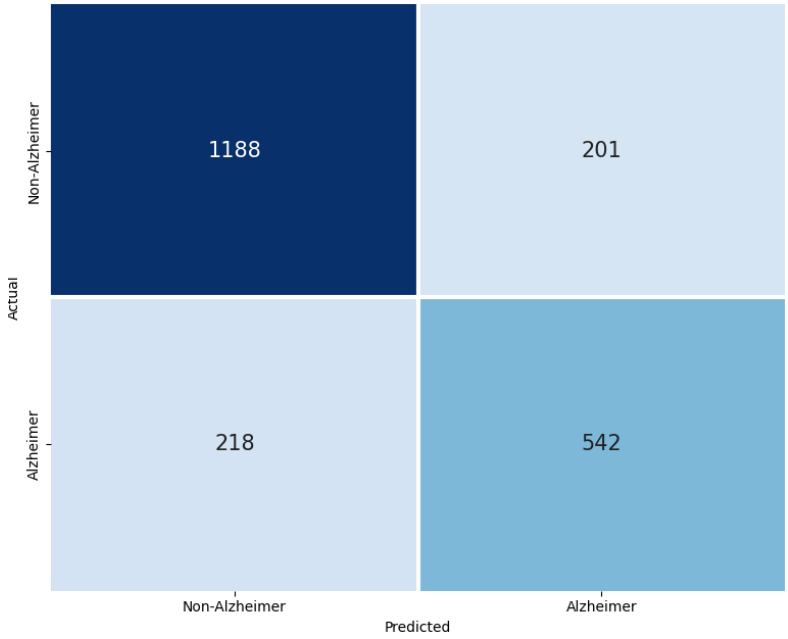


Figure 7. Accumulated confusion matrix of 32-63-1 architecture with learning rate of 0.01 without ADASYN.

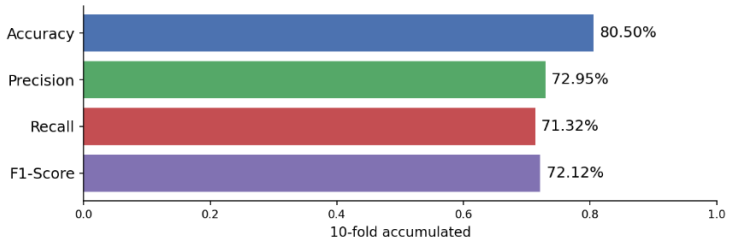


Figure 8. Mean metrics of 32-63-1 architecture with learning rate of 0.01 without ADASYN.

The learning rate and architecture also influenced these results. With ADASYN, a learning rate of 0.09 yielded the lowest mean accuracy (79.59%), whereas learning rates of 0.01 and 0.05 were nearly identical (80.87% and 80.94%, respectively); without ADASYN, a learning rate of 0.09 was also the lowest (80.27%). The mean recall with ADASYN was highest at a learning rate of 0.01 (71.45%), suggesting that a small learning rate is better suited to being combined with synthetic data. In terms of architecture, the mean accuracies of 32-33-1, 32-48-1, and 32-63-1 were 80.02%, 80.24%, and 81.14%, respectively, with ADASYN and 80.39%, 80.27%, and 80.58%, respectively, without ADASYN. The 32-63-1 architecture was highest under both conditions, and the benefit of ADASYN for accuracy was most apparent for this architecture (+0.56 percentage points), whereas for the 32-33-1 architecture, ADASYN lowered the mean accuracy (−0.37 percentage points) and recall (70.53% compared with 73.07%). Increasing the number of hidden neurons from 33 to 48 did not yield an improvement, whereas increasing it to 63 yielded a slight improvement.

Because ADASYN was applied only to the training data and the test data of each fold contained no synthetic data, the results on the test data reflected the model's ability to generalize to new

data. With 10-fold cross-validation, all 2,149 records (760 Alzheimer) served as test data exactly once, so a single prediction in the accumulated confusion matrix changed the accuracy by 0.05 percentage points and recall by 0.13 percentage points. The accuracy difference between the conditions (a maximum of 1.07 percentage points) was smaller than the standard deviation across folds (1.54 to 3.28 percentage points) and therefore may not be meaningful. Accordingly, the most consistent finding of this study is that ADASYN increased precision in six of nine configurations and reduced false alarms, but did not improve recall (higher in two of nine configurations) providing no better results for Alzheimer's disease diagnosis.

4. CONCLUSION

The application of ADASYN to the training data did not consistently improve the accuracy of the BPNN and did not improve the model's ability to recognize the Alzheimer class. On the 10-fold test data, ADASYN produced higher accuracy in four of nine configurations, with a mean of 80.47% compared with 80.41% without ADASYN, whereas recall was higher in only two of nine configurations, with a mean of 70.83% compared with 72.28%. Weaker recall was observed when using ADASYN compared to scenarios without data balancing, providing no better results for Alzheimer's disease diagnosis. The best configuration with ADASYN was the 32-63-1 architecture with a learning rate of 0.01, an accuracy of 81.57%, precision of 75.05%, recall of 71.84%, and F1-score of 73.31%. In the same configuration without ADASYN, accuracy reached 80.50% and recall 71.32%, so ADASYN increased accuracy by 1.07 pp and recall by 0.52 pp and reduced False Negatives from 218 to 214 cases only in this configuration and did not improved classification performance in general. Overall, the benefits of ADASYN are limited and depend on the configuration and dataset. Future research is recommended to study it on external datasets, test a broader range of architectures, learning rates, and numbers of epochs, and compare ADASYN with other balancing techniques.

AUTHOR CONTRIBUTIONS

Alghozi Rafif Ali: Data curation, preprocessing, software implementation, investigation, neural network training, formal analysis, visualization, and writing—original draft. Iis Afrianty: Conceptualization, methodology, research planning, supervision, validation, interpretation of findings, and writing—review and editing. Pizaini Pizaini: Methodological development, software validation, evaluation of model configurations, interpretation of results, and manuscript review. Yusra Yusra: Formal analysis, verification of evaluation metrics, validation of experimental results, and writing—review and editing. Nazruddin Safaat Harahap: Research supervision, methodological evaluation, critical interpretation of findings, and writing—review and editing. All authors contributed to manuscript revision, reviewed and approved the final version, and agreed to be accountable for the accuracy and integrity of the work.

FUNDING STATEMENT

This research received no external funding.

DECLARATION OF AI USE

The authors declare that no AI tools were used in preparing this manuscript.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

All data supporting this study are included in the article. No separate datasets are available.

ETHICAL APPROVAL

Ethical approval was unnecessary, as no humans or animals were involved.

ACKNOWLEDGMENTS

The authors would like to thank the Department of Informatics Engineering, Universitas Islam Negeri Sultan Syarif Kasim Riau for providing academic support. The authors also thank the medical research repositories and data contributors for providing access to the Alzheimer's Disease clinical dataset for research and testing. Finally, the authors are grateful to their colleagues and supervisors for their helpful feedback, suggestions, and guidance throughout the research process.

REFERENCES

[1] El-Assy, A. M., Amer, H. M., Ibrahim, H. M., & Mohamed, M. A. (2024). A novel CNN architecture for accurate early detection and classification of Alzheimer's disease using MRI data. *Scientific Reports*, **14**(1), 3463.
DOI: https://doi.org/10.1038/s41598-024-53733-6

[2] Liu, W., Deng, W., Gong, X., Ou, J., Yu, S., & Chen, S. (2025). Global burden of Alzheimer's disease and other dementias in adults aged 65 years and over, and health inequality related to SDI, 1990–2021: analysis of data from GBD 2021. *BMC Public Health*, **25**(1), 1256.
DOI: https://doi.org/10.1186/s12889-025-22378-z

[3] Nurmawanti, L. & Sudaryanto, A. (2025). Case report: Penatalaksanaan holistik penyakit Alzheimer pada lansia di panti sosial. *J. Penelit. Perawat Prof*, **7**(5474), 1333-1336.
DOI: https://doi.org/10.37287/jppp.v7i1.5783

[4] Setyawati, R., Astuti, A., Utami, T. P., Adiwijaya, S., & Hasyim, D. M. (2024). The importance of early detection in disease management. *Journal of World Future Medicine, Health and Nursing*, **2**(1), 51-63.
DOI: https://doi.org/10.70177/health.v2i1.692

[5] JB, S. N., Sitorus, I. P., & Aprilia, K. T. (2026). Klasifikasi tingkat demensia Alzheimer's berbasis machine learning dan deep learning. *Jurnal Nasional Teknologi Komputer*, **6**(3), 1220-1228.
DOI: https://doi.org/10.61306/jnastek.v6i3.476

[6] Ahmed, G., Er, M. J., Fareed, M. M. S., Zikria, S., Mahmood, S., He, J., Asad, M., Jilani, S. F., & Aslam, M. (2022). Dad-net: Classification of alzheimer's disease using adasyn oversampling technique and optimized neural network. *Molecules*, **27**(20), 7085.
DOI: https://doi.org/10.3390/molecules27207085

[7] Ozsahin, I., Sekeroglu, B., & Mok, G. S. (2019). The use of back propagation neural networks and 18F-Florbetapir PET for early detection of Alzheimer's disease using Alzheimer's Disease Neuroimaging Initiative database. *PLoS One*, **14**(12), e0226577.
DOI: https://doi.org/10.1371/journal.pone.0226577

[8] Wang, N., Chen, J., Xiao, H., Wu, L., Jiang, H., & Zhou, Y. (2019). Application of artificial neural network model in diagnosis of Alzheimer's disease. *BMC Neurology*, **19**(1), 154.
DOI: https://doi.org/10.1186/s12883-019-1377-4

[9] Azhima, M. ., Afrianty, I., Budianita, E. ., & Gusti, S. K. (2024). Penerapan metode backpropagation neural network untuk klasifikasi penyakit stroke. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, **4**(6), 3013-3021.
DOI: https://doi.org/10.30865/klik.v4i6.1956

[10] Wardhana, F. K., Jati, N. S., Seto, B. R., Wati, F. M., Yuliana, Y. A., & Kusumastuti, R. (2024, December). Penerapan backpropagation jaringan saraf tiruan untuk prediksi diabetes menggunakan dataset pima indians. *Prosiding Seminar Nasional Amikom Surakarta*, **2**, 331-344.
URL: https://ojs.amikomsolo.ac.id/index.php/semnasa/article/view/485

[11] Sangalli, S., Erdil, E., Hötker, A., Donati, O., & Konukoglu, E. (2021). Constrained optimization to train neural networks on critical and under-represented classes. *Advances in Neural Information Processing Systems*, **34**, 25400-25411.
URL: https://proceedings.neurips.cc

[12] Bintoro, P., Ratnasari, R., Wihardjo, E., Putri, I. P., & Asari, A. (2024). *Pengantar machine learning*. PT. Mafy Media Literasi Indonesia.
URL: https://repository.um.ac.id/5619/1/fullteks.pdf

[13] Hassoon, I. M. (2025). ADASYN oversampling and SGB ensemble algorithm for migraine classification. *Iraqi Journal of Science*, **66**(2), 788-800.
DOI: https://doi.org/10.24996/ijs.2025.66.2.20

[14] Aulia, S. R. Z., Okfalisa, O., Haerani, E., & Oktavia, L. (2025). Application of ADASYN technique in classification of stroke disease using backpropagation neural network. *INOVTEK Polbeng - Seri Informatika*, **10**(3), 1666-1674.
DOI: https://doi.org/10.35314/jdhv9s39

[15] Sirait, M. T. T. B., Fathonah, N. S., & Fauzan, M. N. (2024). Pemanfaatan algoritma ADASYN dan support vector machine dalam meningkatkan akurasi prediksi kanker paru-Paru. *JATI (Jurnal Mahasiswa Teknik Informatika)*, **8**(5), 8773-8778.
DOI: https://doi.org/10.36040/jati.v8i5.10752

[16] Ramadhanti, D. V., Santoso, R., & Widiharih, T. (2023). Perbandingan SMOTE dan ADASYN pada data imbalance untuk klasifikasi rumah tangga miskin di Kabupaten Temanggung dengan algoritma K-nearest neighbor. *J. Gaussian*, **11**(4), 499-505.
DOI: https://doi.org/10.14710/j.gauss.11.4.499-505

[17] Gress, T. W., Denvir, J., & Shapiro, J. I. (2018). Effect of removing outliers on statistical inference: implications to interpretation of experimental data in medical research. *Marshall Journal of Medicine*, **4**(2), 9.
DOI: https://doi.org/10.33470/2379-9536.1170

[18] Alfarizi, N., Sinurat, S., Putra, A., Amin, M., Lydia, P., & Budi, U. P. (2026). Analisis penanganan data tidak seimbang terhadap kinerja klasifikasi sentimen multikelas pada ulasan marketplace Tokopedia. *Journal of Science and Social Research*, **9**(1), 474-482.
DOI: https://doi.org/10.54314/jssr.v9i1.5804

[19] Chi, Q., Yang, Z., & Liang, H. P. (2022). A transendothelial leukocyte transmigration model based on computational fluid dynamics and BP neural network. *Frontiers in Bioengineering And Biotechnology*, **10**, 881797.
DOI: https://doi.org/10.3389/fbioe.2022.881797

[20] Saluza, I., Astuti, L. W., & Yulianti, E. (2023). Ensemble backpropagation neural network dalam memprediksi inflasi. *J. JUPITER*, **15**(1), 732-741.
DOI: https://doi.org/10.5281./6613/15.jupiter.2023.04

[21] Hovi, H. S. W., Hadiana, A. I., & Umbara, F. R. (2022). Prediksi Penyakit Diabetes Menggunakan Algoritma Support Vector Machine (SVM). *Informatics and Digital Expert (INDEX)*, **4**(1), 40-45.
DOI: https://doi.org/10.36423/index.v4i1.895

[22] Aziz, M. I. (2025). Implementasi machine learning untuk deteksi anomali kesehatan janin menggunakan metode ensemble berbasis decision tree. *Jurnal Tecnoscienza*, **9**(2), 189-198.
DOI: https://doi.org/10.51158/zymd0m16

[23] Romadloni, P., Kusuma, B. A., & Baihaqi, W. M. (2022). Komparasi metode pembelajaran mesin untuk implementasi pengambilan keputusan dalam menentukan promosi jabatan karyawan. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), 622-628.
DOI: https://doi.org/10.36040/jati.v6i2.5238