Volume XI Issue 1 Year 2026 | Page 331-342 ISSN: 2527-9866
*Received: **27-01-2026** | Revised: **10-02-2026**| Accepted:**25-02-2026***

Alzheimer's Disease Classification Using the Tabnet Model Enhanced by Hyperparameter Optimization

Muchamad Bagus Triyadi[1], Christian Sri Kusuma Aditya[2]
[1,2] University of Muhammadiyah Malang, Malang, East Java, Indonesia, 65144
e-mail: muchbagus17@webmail.umm.ac.id[1], christianskaditya@umm.ac.id[2]
*Correspondence: muchbagus17@webmail.umm.ac.id

Abstract: *Alzheimer's disease is a progressive neurodegenerative disorder that leads to a gradual decline in cognitive function and remains challenging to diagnose at an early stage, as clinical symptoms often emerge after substantial brain damage has occurred. Therefore, accurate and efficient predictive models based on clinical data are essential to support early detection. Recent advances in deep learning for tabular data, particularly the TabNet model, enable adaptive feature selection through attention mechanisms while preserving interpretability. This study applies TabNet for Alzheimer's disease classification using clinical tabular data and enhances its performance through hyperparameter optimization employing Grid Search, Random Search, and Bayesian Optimization. Model evaluation was conducted using accuracy, area under the curve (AUC), confusion matrix analysis, and execution time. Experimental results show that Random Search achieved the highest classification accuracy of 90.53%, whereas Bayesian Optimization obtained the highest AUC of 94.82%, indicating superior discriminative capability. These results demonstrate that integrating TabNet with appropriate hyperparameter optimization strategies provides a competitive, efficient, and interpretable approach for Alzheimer's disease classification, supporting its potential application in data-driven clinical decision support systems.*

Keywords: *Alzheimer's disease, early detection, deep learning, tabular data, TabNet.*

1. Introduction

Alzheimer's disease is a progressive neurodegenerative condition marked by amyloid plaque accumulation, tau pathology, and neuronal degeneration, leading to gradual declines in cognitive function, behavior, and social abilities [1] Early diagnosis remains challenging because clinical symptoms typically appear only after substantial brain damage has occurred. While advances in neuroimaging techniques, such as positron emission tomography (PET), and plasma biomarkers have improved the detection of Alzheimer's pathology, their high costs and limited accessibility restrict widespread clinical adoption. Therefore, the development of efficient and accessible predictive methods is urgently needed. Alzheimer's disease is the primary cause of dementia worldwide, with its prevalence increasing in parallel with the aging population. Epidemiological data indicate that the global prevalence of dementia in 2019 is expected to nearly double by 2050, with Alzheimer's disease accounting for approximately 60–70% of cases [2]. This growing trend poses significant social, economic, and healthcare challenges, underscoring the need for early detection and more accurate classification methods to support clinical decision-making.

In the United States, more than 6.2 million individuals aged 65 years and older were living with Alzheimer's disease in 2021, with a higher prevalence observed among women compared to men [3]. As prevalence rates continue to increase, early diagnosis is widely regarded as a critical strategy for slowing disease progression, enabling earlier intervention, and reducing the psychological and economic burden experienced by patients' families [4].

With the advancement of deep learning for tabular data, models such as TabNet and TabTransformer have emerged as adaptive approaches for utilizing clinical and demographic information. TabNet applies attention mechanisms to emphasize relevant features while maintaining interpretability, which is crucial for medical applications, whereas TabTransformer uses transformer-based contextual representations to capture complex feature interactions [5]. Both models show strong potential to enhance the accuracy and reliability of tabular data classification, including for Alzheimer's disease diagnosis.

2. Literature Review

Recent studies have highlighted the challenges of early Alzheimer's disease diagnosis. Although imaging techniques and biomarkers such as amyloid PET, plasma biomarkers, MRI, PET, and EEG can detect pathological changes before clinical symptoms appear, their high cost, limited accessibility, and variability in patient responses hinder widespread clinical adoption [6]. TabNet was first introduced by Arik and Pfister in 2019 as a deep learning model specifically designed for tabular data [7]. It employs a sequential attention mechanism to focus on relevant features at each decision step, improving interpretability compared to traditional models such as XGBoost and Random Forest while maintaining competitive performance. The use of sparse attention further enhances computational efficiency, making TabNet well suited for healthcare applications that require transparency, efficiency, and adaptability across domains, including bias-aware modeling. Grid Search is a classical hyperparameter optimization technique that systematically evaluates all possible parameter combinations within a predefined search space, making it a common baseline due to its simplicity and reproducibility [8]. However, as the number of hyperparameters increases, its computational cost grows exponentially, rendering Grid Search inefficient for models with large and complex search spaces.

Random Search is a widely used hyperparameter optimization method that explores the search space through random sampling to identify effective parameter configurations. Prior studies proposed an agent-based collaborative random search that improves efficiency by combining global exploration with cooperative agents [9] while Weighted Random Search assigns probabilistic weights to parameter distributions to increase the likelihood of finding optimal configurations within a shorter time [10]. These approaches demonstrate that Random Search can be enhanced through collaborative and weighted strategies to improve its optimization effectiveness.

Bayesian Optimization is a widely adopted hyperparameter tuning method that utilizes probabilistic modeling to efficiently identify optimal parameter configurations. Unlike Grid Search and Random Search, it employs a surrogate model to iteratively approximate the objective function, allowing more focused exploration of the search space. Bayesian Optimization can accelerate the hyperparameter tuning process while improving model accuracy by efficiently exploring the search space using a probabilistic framework [11] report that Bayesian Optimization accelerates the tuning process while improving model accuracy, while [12] highlights its effectiveness in handling complex hyperparameter spaces in neural networks. Accuracy is a commonly used evaluation metric that reflects the proportion of correct predictions; however, it can be misleading in the presence of class imbalance, [13] o overcome this limitation, many studies employ the Area Under the Curve (AUC), which provides a more balanced assessment of model performance and is more suitable for model selection, [14] . AUC effectively measures a model's ability to distinguish between classes regardless of class prevalence [15] Therefore, combining accuracy and AUC offers a more robust and fair evaluation framework for predictive models.

3. Methods

A. Research Workflow

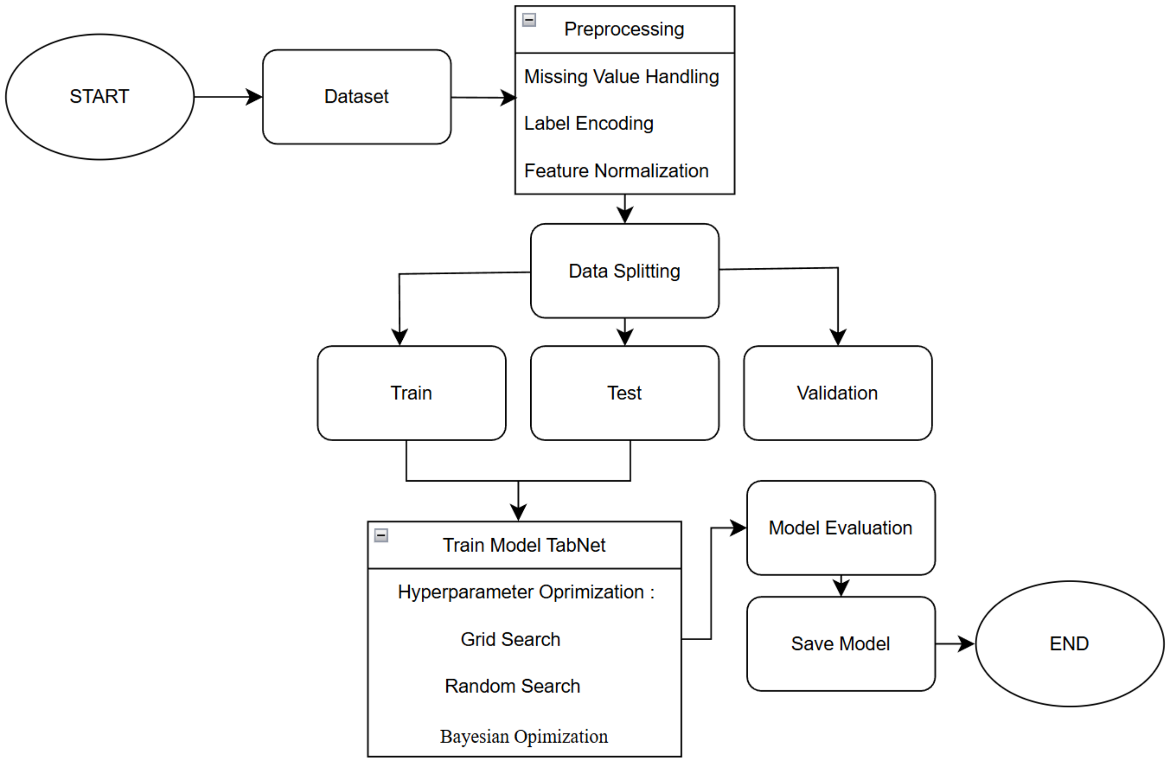

Figure 1. Research Workflow

The research workflow begins with dataset collection and preprocessing, including missing value handling, label encoding, and feature normalization, followed by data splitting into training, validation, and testing sets. The TabNet model is then trained using Grid Search, Random Search, and Bayesian Optimization to identify optimal hyperparameter configurations, with model selection based on validation performance. Finally, the selected models are evaluated on the test set using metrics such as accuracy, execution time, hyperparameter analysis, confusion matrix, and area under the curve (AUC) to assess the effectiveness of TabNet with hyperparameter optimization.

B. Data Description

This study uses an open-access Alzheimer's Disease Dataset from Kaggle comprising tabular clinical data, including demographic information and cognitive assessment scores. The dataset's structured format and representative diagnostic features support effective deep learning–based classification, as reported in [16], An overview of the dataset attributes used in this study is provided in Table I. Sample Data**.**

The dataset consists of a total of 2,149 samples, categorized into two classes based on the diagnosis attribute, which serves as the target label for Alzheimer's disease classification. The class distribution is imbalanced, with approximately 64.63% non-Alzheimer's cases and 35.37% Alzheimer's cases, reflecting realistic clinical conditions. Table I presents a sample of the dataset, illustrating variability across features such as cognitive assessment scores, body mass index, physical activity, and cognitive–behavioral indicators. The dataset contains 35 attributes covering demographic, lifestyle, and clinical variables. Most features are continuous numerical variables, while others are binary or ordinal categorical variables, providing a structured representation suitable for deep learning–based Alzheimer's disease classification.

Table 2. Sample Data

Attribute	Sample 1	Sample 2
patient_id	4751	4752
age	73	89
gender	0	0
ethnicity	0	0
education_level	2	0
bmi	22.927	26.827
smoking	0	0
alcohol_consumption	13.297	4.542
physical_activity	6.327	7.619
diet_quality	1.347	0.518
sleep_quality	9.025	7.151
family_history_alzheimers	0	0
cardiovascular_disease	0	0
diabetes	1	0
depression	1	0
head_injury	0	0
hypertension	0	0
systolic_bp	142	115
diastolic_bp	72	64
cholesterol_total	242	231.162
cholesterol_ldl	56.150	193.407
cholesterol_hdl	33.682	79.028
cholesterol_triglycerides	162.189	294.630
mini_mental_state_exam	21.463	20.613
functional_assessment	6.518	7.118
memory_complaints	0	0
behavior_problems	0	0
activities_of_daily_living	1.725	1.592
confusion	0	0
disorientation	0	0
personality_changes	0	0
difficulty_completing_task	1	0
forgetfulness	0	1
diagnosis	0	0

C. Preprocessing

After obtaining the Alzheimer’s Disease Dataset from Kaggle, preprocessing was performed to prepare the data for TabNet training. This process included handling missing values by converting infinite values to NaN, removing incomplete records, and applying mean and mode imputation for numerical and categorical features, respectively. Categorical variables were encoded using LabelEncoder, numerical features were normalized with StandardScaler, and irrelevant attributes such as *PatientID* and *DoctorInCharge* were removed to ensure suitability for classification model training [17].

1) Missing Value Handling
   In this stage, missing and infinite values were identified, with infinite values converted to NaN for consistent handling. Data imputation was applied by replacing missing numerical values with the column mean and categorical values with the mode, ensuring data completeness and minimizing bias prior to encoding and normalization.
2) Label Encoding

At this stage, categorical features (gender, familyhistoryalzheimers, and diagnosis) were transformed into numerical representations using Label Encoding, with the diagnosis attribute defined as the target label. This encoding ensures compatibility with the TabNet model while preserving the semantic information of categorical variables.

3) Feature Normalization
Numerical features, including age, systolic and diastolic blood pressure, and BMI, were normalized using StandardScaler to achieve zero mean and unit variance. This normalization prevents scale dominance among features, supports stable model convergence, and reduces scale-induced bias, thereby preparing the data for effective TabNet training.

D. Data Splitting

After preprocessing, the dataset was split into training, validation, and test sets using a stratified split to preserve the original class distribution. The data were divided with a ratio of 70% for training, 15% for validation, and 15% for testing. A fixed random seed was applied to ensure reproducibility of the experimental results. The training set was used for model learning, the validation set for hyperparameter optimization, and the independent test set for final performance evaluation.

E. Model TabNet

The primary model employed in this study is TabNet, a deep learning architecture specifically designed for tabular data. TabNet was first introduced in 2019 by Arik and Pfister [7] as an interpretable neural network that leverages a sequential attention mechanism to adaptively select the most relevant features at each decision step. Through this mechanism, the model processes tabular features in a stepwise manner, allowing it to focus on informative attributes while suppressing less relevant ones.

A key advantage of TabNet is its interpretability, as attention masks reveal feature importance at each decision step. This capability is especially valuable in medical applications that require transparent and explainable models for clinical decision support. By combining strong predictive performance with feature-level interpretability, TabNet provides a robust framework for classification on structured clinical data. The mathematical formulation of the TabNet architecture is presented in in Equation (1).

$$M^{(i)} \equiv Sparsemax\ (P^{(i-1)}.\, h\ (a^{(i-1)}))$$

where $M^{(i)}$ denotes the feature selection mask at the i-th decision step, determining which features are emphasized during the current stage. The prior scale vector $P^{(i-1)}$ controls feature reuse by penalizing features selected in previous steps, while $a^{(i-1)}$ represents the attention transformer output that provides contextual information for feature selection. The function $h(\cdot)$denotes a fully connected transformation used to compute feature importance scores. Sparsemax is applied as a normalization function to produce sparse attention weights, enabling the model to focus on the most relevant features while suppressing less important ones.

F. Hyperparameter Optimization

Hyperparameter optimization was conducted to improve the performance and generalization capability of the TabNet model using three strategies: Grid Search, Random Search, and Bayesian Optimization, which were evaluated under identical experimental conditions. All optimization procedures were performed on the validation set, while the training set was used for model learning and the test set was reserved exclusively for final performance evaluation. The hyperparameter search space was defined consistently across all methods to ensure a fair comparison, including the decision and attention dimensions (*n_d* and *n_a*), the number of decision steps (*n_steps*), the attention relaxation parameter (*gamma*), and the sparsity regularization coefficient (*lambda_sparse*). Specifically, *n_d* and *n_a* were selected from {8, 16, 24}, *n_steps* from {3, 5, 6}, *gamma* from {1.0, 1.3, 1.5}, and *lambda_sparse* from {0.0001, 0.001, 0.0013}.

1) Grid Search
This method exhaustively evaluates all hyperparameter combinations within a predefined search space. While Grid Search ensures comprehensive exploration, it is computationally expensive and time-inefficient for large search spaces [18]. The formulation of the Grid Search evaluation is presented in Equation (2).

$$\theta^* = \arg\max_{\theta \sim \mathcal{G}} Score(\theta)$$

where θ^* denotes the optimal hyperparameter configuration obtained through Grid Search. The symbol θ represents a candidate hyperparameter set evaluated during the search, while $\mathcal{G}$ denotes the predefined search space containing all possible parameter combinations. The function $\text{Score}(\theta)$ refers to the evaluation metric, such as accuracy or AUC, computed on the validation dataset. Grid Search selects the configuration that maximizes this evaluation score over the entire search space.

2) Random Search
Unlike Grid Search, Random Search randomly samples hyperparameter configurations from a predefined search space, making it more efficient when only a subset of parameters significantly influences model performance by avoiding exhaustive evaluation of less impactful combinations. [19]. The formulation for hyperparameter evaluation using the Random Search method is presented in Equation (3).

$$\theta^* = \arg\max_{\theta \sim \mathcal{D}} Score(\theta)$$

Where θ^*denotes the optimal hyperparameter configuration obtained through Random Search. The symbol θ represents a hyperparameter set randomly sampled from the predefined distribution $\mathcal{D}$ and $\text{Score}(\theta)$ denotes the evaluation metric computed on the validation dataset. By maximizing this score over randomly sampled configurations, Random Search enables more efficient exploration of the search space when only a subset of hyperparameters has a significant impact on model performance.

3) Bayesian Optimization
Bayesian Optimization employs a probabilistic surrogate model to iteratively select promising hyperparameter configurations by balancing exploration and exploitation, enabling efficient identification of optimal settings with fewer evaluations compared to exhaustive or random search methods [20]. The formulation for hyperparameter optimization using the Bayesian Optimization method is presented in Equation (4).

$$\theta^* = arg\max_{\theta} \mu(\theta) + \mathcal{K}\sigma(\theta)$$

where θ^*denotes the optimal hyperparameter configuration obtained through Bayesian Optimization. The function $\mu(\theta)$represents the predictive mean of the surrogate model, estimating the expected performance of a given configuration, while $\sigma(\theta)$denotes the associated predictive uncertainty. The parameter κ controls the trade-off between exploration and and exploitation by weighting the uncertainty term in the acquisition function. By maximizing $\mu(\theta) + \kappa\sigma(\theta)$, Bayesian Optimization efficiently identifies promising configurations while exploring uncertain regions of the search space.

All experiments were conducted in Python using the PyTorch framework and the PyTorch-TabNet library within a Jupyter Notebook environment, with training performed on an NVIDIA RTX 3060 GPU. Each TabNet model was trained for up to 50 epochs with early stopping based on AUC to mitigate overfitting. Hyperparameter optimization employed a stratified training–validation split to address class imbalance, and model selection was based on validation AUC. Grid Search evaluated 32 configurations using 3-fold cross-validation (96 fits), Random Search explored 10 configurations (30 fits), and Bayesian Optimization applied an iterative probabilistic strategy under identical training settings to ensure fair comparison.

G. Model Evaluation

Model evaluation was performed to assess the classification performance of the TabNet model optimized using Grid Search, Random Search, and Bayesian Optimization. Multiple evaluation metrics, including AUC, confusion matrix, execution time, and hyperparameter analysis, were employed to provide a comprehensive comparison in terms of predictive performance, reliability, and computational efficiency. This multi-metric evaluation ensures a robust and practical assessment of each optimized model.

1) AUC (Area Under Curve)
The primary evaluation metric in this study is the Area Under the Curve (AUC), which measures the balance between sensitivity and specificity. AUC is particularly suitable for binary classification tasks with imbalanced datasets, as commonly encountered in medical applications such as Alzheimer’s disease classification. Higher AUC values indicate stronger discriminative capability between classes [21] .

2) Confusion Matrix
A confusion matrix is used to provide detailed insight into model predictions relative to the true class labels. It summarizes correct and incorrect classifications for each class, enabling the calculation of metrics such as precision, recall, and F1-score, and supports analysis of misclassification patterns and potential class bias, which is particularly important in medical classification tasks [22].

3) Execution Time Analysis of Hyperparameter Optimization Strategies
In addition to predictive performance, execution time is evaluated as a key indicator of computational efficiency for real-world deployment. Accordingly, both training and inference times are measured for each optimized model to assess their computational performance [23].

4) Evaluation of Hyperparameter Configurations on TabNet Performance
This evaluation analyzes the impact of hyperparameter configurations on TabNet performance using Grid Search, Random Search, and Bayesian Optimization. Key parameters include n_d and n_a (decision and attention dimensions), n_steps (decision depth), gamma (attention relaxation), and lambda_sparse (sparsity regularization). which influence model performance, complexity, and interpretability in Alzheimer’s disease classification [24].

4. Results And Discussion

A. Performance Evaluation Based on Accuracy, AUC, and Execution Time

Hyperparameter tuning is crucial for optimizing machine learning model performance. In this study, Grid Search, Random Search, and Bayesian Optimization were applied to the TabNet model for Alzheimer's disease classification, each employing different search strategies to identify optimal configurations. The comparative performance of these methods in terms of accuracy and AUC is summarized in Table 2.

Table 2. Classification Performance on the Test Set

Model Classification	Accuracy (%)	Class	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
TabNet + Grid Search	83,02	0	86,00	87,00	87.00	87.19
		1	77,00	75,00	76.00	
TabNet + Random Search	90,53	0	90,00	97.00	93.00	93.75
		1	93,00	80.00	86.00	
TabNet + Bayesian Optimization	91,63	0	91,00	96.00	94.00	94.82
		1	93,00	83.00	88.00	

Table 2 summarizes the classification performance of TabNet with different hyperparameter optimization strategies on the test set, evaluated using accuracy, precision, recall, F1-score, and AUC. In addition to predictive performance, computational efficiency was assessed by measuring the execution time of each tuning method three times, with the average execution time per epoch reported in Table 3.

Table 3. Average Execution Time per Epoch (Seconds)

Model Classification	Grid Search (S)	Random Search (S)	Bayesian Optimization (S)
Tabular Network (TabNet)	944.25	845.62	835.83

As shown in Table 3, Grid Search consistently records the highest execution time across all experiments due to its exhaustive exploration of the hyperparameter space. In contrast, Bayesian Optimization achieves the lowest execution time, reflecting its probabilistic and targeted search strategy. Random Search demonstrates intermediate performance, with execution times lower than Grid Search but higher than Bayesian Optimization. Overall, these results indicate that Bayesian Optimization offers a more favorable balance between computational efficiency and optimization effectiveness.

B. Comparison of Results Based on the Confusion Matrix

Model performance was evaluated using confusion matrix analysis to compare the classification capability of the TabNet model under Grid Search, Random Search, and Bayesian Optimization based on true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values [25]. Grid Search produced TP = 242, FN = 35, TN = 115, and FP = 38, indicating moderate performance with a relatively higher false negative rate. Random Search improved detection capability with TP = 268, FN = 9, TN = 123, and FP = 30. Bayesian Optimization yielded the most balanced results (TP = 267, FN = 10, TN = 127, FP = 26), demonstrating robust performance across both classes. The corresponding confusion matrices are presented in Figures 2, 3, and 4.

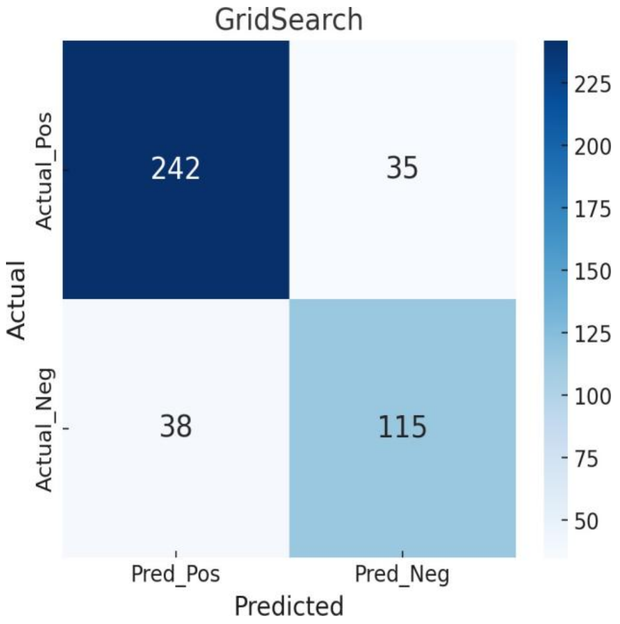

Figure 2. Confusion Matrix Grid Search

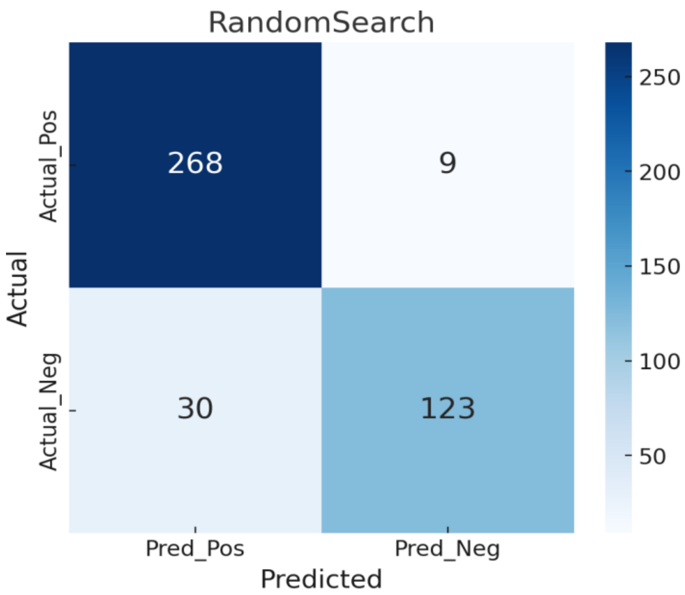

Figure 3. Confusion Matrix Random Search

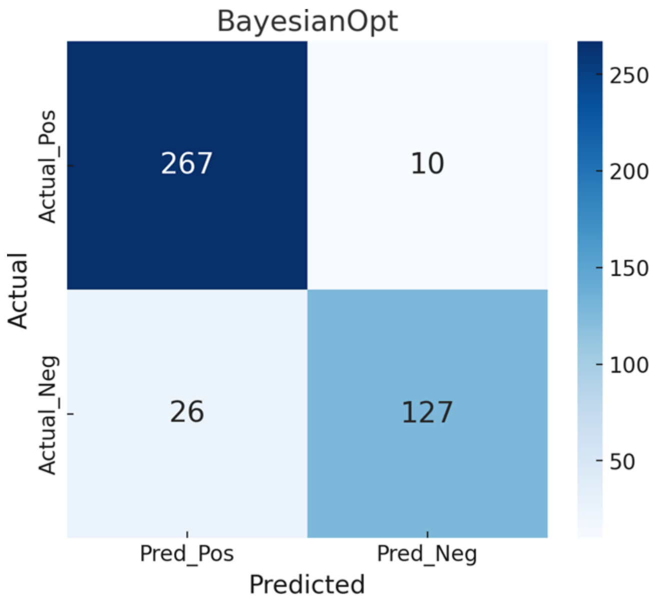

Figure 4. Confusion Matrix Bayesian Optimization

C. Evaluation of Parameter Combinations on Model Performance

This study evaluated optimal TabNet hyperparameter configurations using Grid Search, Random Search, and Bayesian Optimization, focusing on key parameters including n_d, n_a, n_steps, gamma, and lambda_sparse, which influence model capacity, sparsity, and generalization performance. Each tuning strategy identified a distinct optimal configuration, which was subsequently retrained and evaluated on the independent test set. The resulting parameter combinations and corresponding test performance metrics are summarized in Table 4.

Table 4. Performance Results on the Test Set Based on Parameter Combinations

Method Tuning	n_d	n_a	n_steps	Gamma	Lambda _sparse	AUC (%)	Accuracy (%)
Grid Search	8	8	3	1.3	0.0001	87.19	83.02
Random Search	24	24	5	1.0	0.001	93.75	90,53
Bayesian Optimization	16	16	6	1.5	0.0013	94.82	91.63

As shown in Table 4, each hyperparameter optimization method demonstrates competitive performance with distinct strengths. Grid Search provides stable results, Random Search achieves the highest accuracy, and Bayesian Optimization attains the highest AUC, reflecting superior discriminative capability. These findings indicate that optimal hyperparameter selection should be guided by task-specific priorities, such as accuracy, AUC, or computational efficiency.

D. Discussion

The results indicate that TabNet optimized with Random Search achieved the highest accuracy of 90.53%, outperforming Grid Search and Bayesian Optimization under identical experimental conditions, while Bayesian Optimization produced the highest AUC of 94.82%, reflecting stronger discriminative capability and generalization performance, particularly for imbalanced data. These findings demonstrate that different hyperparameter optimization strategies emphasize distinct performance objectives, such as overall classification accuracy or discriminative power. From a practical perspective, Random Search may be more suitable for applications prioritizing classification correctness, whereas Bayesian Optimization is advantageous when robust class discrimination is required in imbalanced clinical datasets. Overall, TabNet exhibits strong classification performance, computational efficiency, and interpretability through attentive feature selection, supporting its applicability as a reliable and flexible approach for tabular medical data analysis and clinical decision support.

5. Conclusion

This study demonstrates that TabNet is an effective model for Alzheimer's disease classification using tabular clinical data. Random Search achieved the highest accuracy (90.53%), while Bayesian Optimization produced the highest AUC (94.82%), indicating that different hyperparameter tuning strategies emphasize distinct performance objectives. The results further show that careful hyperparameter optimization plays a critical role in balancing predictive performance, generalization capability, and computational efficiency. In addition, the attention-based architecture of TabNet enhances model interpretability by enabling feature-level insight, which is particularly valuable in clinical decision-making contexts. Overall, the integration of TabNet with hyperparameter optimization provides a competitive and flexible approach for Alzheimer's disease prediction, supporting its applicability as a reliable decision-support tool for data-driven medical diagnosis and future clinical applications.

References

[1] P. Scheltens, B. De Strooper, M. Kivipelto, H. Holstege, G. Chételat, and C. E. Teunissen, "Alzheimer's disease," The Lancet, vol. 397, no. 10284, pp. 1577–1590, 2021.

[2] E. Nichols, J. D. Steinmetz, S. E. Vollset, K. Fukutaki, J. Chalek, F. Abd-Allah, et al., "Estimation of the global prevalence of dementia in 2019 and forecasted prevalence in 2050: An analysis for the Global Burden of Disease Study 2019," The Lancet Public Health, vol. 7, no. 2, pp. e105–e125, 2022.

[3] N. B. Balakrishnan, P. S. Sreeja, and J. J. Panackal, "Alzheimer's disease diagnosis using machine learning: A review," arXiv preprint arXiv:2304.09178, 2023.

[4] Foroughipoor, K. Moradi, and H. Bolhasani, "Alzheimer's disease diagnosis by deep learning using MRI-based approaches," arXiv preprint arXiv:2310.17755, 2023.

[5] S. V. Razavi-Termeh, A. Sadeghi-Niaraki, A. Sorooshian, T. Abuhmed, and S. M. Choi, "Spatial mapping of land susceptibility to dust emissions using optimization of attentive interpretable tabular learning (TabNet) model," Journal of Environmental Management, vol. 358, p. 120682, 2024.

[6] S. Boughorbel, F. Jarray, and A. Kadri, "Fairness in TabNet model by disentangled representation for the prediction of hospital no-show," arXiv preprint arXiv:2103.04048, 2021.

[7] S. Ö. Arik and T. Pfister, "TabNet: Attentive interpretable tabular learning," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 8, pp. 6679–6687, 2021.

[8] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," Neurocomputing, vol. 415, pp. 295–316, 2020.

[9] A. Esmaeili, Z. Ghorrati, and E. T. Matson, "Agent-based collaborative random search for hyperparameter tuning and global function optimization," Systems, vol. 11, no. 5, p. 228, 2023.

[10] A. C. Florea and R. Andonie, "Weighted random search for hyperparameter optimization," arXiv preprint arXiv:2004.01628, 2020.

[11] A. H. Victoria and G. Maragatham, "Automatic tuning of hyperparameters using Bayesian optimization," Evolving Systems, vol. 12, no. 1, pp. 217–223, 2021.

[12] G. Onorato, "Bayesian optimization for hyperparameters tuning in neural networks," arXiv preprint arXiv:2410.21886, 2024.

[13] L. Ferrer, "Analysis and comparison of classification metrics," arXiv preprint arXiv:2209.05355, 2022.

[14] A. M. Carrington, D. G. Manuel, P. W. Fieguth, T. Ramsay, V. Osmani, and B. Wernly, "Deep ROC analysis and AUC as balanced average accuracy to improve model selection, understanding and interpretation," arXiv preprint arXiv:2103.11357, 2021.

[15] M. S. Rahman and P. G. Prasad, “A comprehensive study on machine learning methods to increase the prediction accuracy of classifiers and reduce the number of medical tests required to diagnose Alzheimer’s disease,” arXiv preprint arXiv:2212.00414, 2022.
[16] S. B. S. Meraj and B. S. Shetty, “Successful data mining: With dimension reduction,” Proceedings of the International Conference on Applied Machine Intelligence and Data Analytics (ICAMIDA), pp. 11–22, 2023.
[17] O. M. Katipoglu, M. A. Pekin, and S. Akil, “The impact of preprocessing approaches on neural network performance: A case study on evaporation in Adana, a Mediterranean climate,” Indonesian Journal of Earth Sciences, vol. 3, no. 2, p. A821, 2023.
[18] M. Ogunsanya, J. Isichei, and S. Desai, “Grid search hyperparameter tuning in additive manufacturing processes,” Manufacturing Letters, vol. 35, pp. 1031–1042, 2023.
[19] R. Andonie and A. C. Florea, “Weighted random search for CNN hyperparameter optimization,” arXiv preprint arXiv:2003.13300, 2020.
[20] [20] S. Malakouti, M. Menhaj, and A. Suratgar, “Applying grid search, random search, Bayesian optimization, genetic algorithm, and particle swarm optimization to fine-tune the hyperparameters of the ensemble of ML models enhances its predictive accuracy for mud loss,” Research Square, preprint, Dec. 2024, doi: 10.21203/rs.3.rs-5187887/v1.
[21] Q. M. Zhou, L. Zhe, R. J. Brooke, M. M. Hudson, and Y. Yuan, “A relationship between the incremental values of area under the ROC curve and of area under the precision–recall curve,” Diagnostic and Prognostic Research, vol. 5, no. 1, p. 13, 2021.
[22] A. Tharwat, “Classification assessment methods,” Applied Computing and Informatics, vol. 17, no. 1, pp. 168–192, 2021.
[23] I. C. Jurado, P. Gijsbers, and J. Vanschoren, “AutoML benchmark with shorter time constraints and early stopping,” arXiv preprint arXiv:2504.01222, 2025.
[24] K. Eggensperger, P. Müller, N. Mallik, M. Feurer, R. Sass, A. Klein, et al., “HPOBench: A collection of reproducible multi-fidelity benchmark problems for hyperparameter optimization,” arXiv preprint arXiv:2109.06716, 2021.
[25] B. Nguyen and Y. Zhang, “A comparative study of diabetes prediction based on lifestyle factors using machine learning,” arXiv preprint arXiv:2503.04137, 2025.