Journal of Engineering and Technology for Industrial Applications

ITEGAM-JETIA

Manaus, v.12 n.57, p. 1148-1161. January/February, 2026.
DOI: https://doi.org/10.5935/jetia. v12i57.3295

ISSN ONLINE: 2447-0228

RESEARCH ARTICLE OPEN ACCESS

RANDOMIZED DENDRITIC NEURON MODEL PRESERVING BIOLOGICALLY INSPIRED DENDRITIC GATING FOR EFFICIENT MEDICAL DIAGNOSTICS

Michael Angello Qadosy Riyadi*[1] and Adinda Mariasti Dewi[2]

[1, 2]Department of Information Technology, Telkom University, Surabaya Campus, Surabaya, Indonesia.

[1]https://orcid.org/0009-0003-0853-0989, [2]https://orcid.org/0009-0008-1157-0478

Email: *michaelangello@student.telkomuniversity.ac.id, adindamariasti@student.telkomuniversity.ac.id

ARTICLE INFO

Article History
Received: January 12, 2025
Revised: January 20, 2026
Accepted: January 30, 2026
Published: February 28, 2026

Keywords:
Dendritic Neuron Model,
Neuromorphic Classifier,
Non-communicable Diseases,
Computationally Efficient,
Medical.

ABSTRACT

Non-communicable diseases (NCDs) such as type 2 diabetes, breast cancer, and cardiovascular diseases remain major global health burdens, requiring accurate yet computationally efficient predictive models. This study aims to develop a fast and scalable neuromorphic classifier for NCD prediction in resource-constrained clinical settings. We propose the Randomized Dendritic Neuron Model (R-DNM), a randomized variant of the Dendritic Neuron Model that preserves multiplicative dendritic interactions while fixing randomly initialized hidden-layer parameters and training only output weights via ridge regression. R-DNM was evaluated on three benchmark medical datasets using 10-fold cross-validation and compared against conventional DNM, its multi-output variant (MODN), and standard baselines including Random Forest, SVM, and MLP. On the Pima Indians Diabetes dataset, R-DNM achieved the best performance with an average accuracy of 0.7552 ± 0.0362, F1-score of 0.7529 ± 0.0343, and log-loss of 0.5668 ± 0.0218, outperforming DNM and MODN by up to 3.38% accuracy. Paired t-tests confirm significant improvements in F1-score ($t = -3.77$, $p = 0.0014$, Cohen's $d = 1.69$) and computation time ($p < 0.0001$). These results establish R-DNM as a computationally efficient and statistically superior neuromorphic model for NCD prediction.

I. INTRODUCTION

Non-communicable diseases (NCDs) such as type 2 diabetes mellitus, breast cancer, and cardiovascular diseases (CVDs) remain leading contributors to global mortality and disease burden. The 11th edition of the International Diabetes Federation (IDF) Diabetes Atlas reports that in 2024, 589 million adults aged 20–79 (11.1%) were living with diabetes, including 252 million undiagnosed cases, with prevalence projected to rise further by 2050 [1-3]. Breast cancer is the most common cancer among women worldwide, with global projections estimating annual incidence to reach approximately 3.2 million cases and 1.1 million deaths by 2050 if current trends continue [4-6]. CVDs remain the leading cause of death globally, with deaths projected to increase from about 20.5 million in 2025 to 35.6 million by 2050 due to population growth and ageing [7], [8]. These trends underscore the urgent need for prevention and early detection.

The integration of data-driven approaches, such as predictive modelling and risk stratification, into healthcare systems—particularly in resource-limited settings—can facilitate earlier diagnosis, targeted prevention, and more effective disease management, ultimately reducing morbidity, mortality, and economic burden [9], [10]. Advancements in machine learning and deep learning have produced various high-performing classification models on benchmark medical datasets, including the Pima Indians Diabetes Database (PIDD), Breast Cancer Wisconsin (Diagnostic), and Cardiovascular Disease Dataset [11-15]. On PIDD, studies have reported accuracies around 0.7710 using ensemble approaches [16]. On the Breast Cancer Wisconsin dataset, ensemble deep learning methods have achieved accuracies up to approximately 0.9736 [17]. For cardiovascular datasets, deep learning models with attention mechanisms have reported accuracies around 0.7552 with interpretability via SHAP [18].

Although these results demonstrate significant progress, most approaches rely on large-scale iterative optimization, posing challenges such as high computational cost, prolonged training time, substantial memory requirements, and vulnerability to vanishing/exploding gradients and overfitting on limited-sized datasets or imbalanced class distributions [19], [20]. The Dendritic Neuron Model (DNM) is an artificial neuron model designed by adopting biological principles of information processing in neurons, particularly the role of dendrites in nonlinear operations [21-23]. Unlike conventional perceptron models that apply only linear aggregation followed by simple nonlinear activation, DNM incorporates four main computational stages: synapses, linear dendritic branches, multiplicative membrane gating, and soma [24-26].

The dendritic gating mechanism enables more selective nonlinear modulation of inputs, allowing the model to capture complex feature interactions in a manner closer to biological processes [27], [28]. Although DNM has shown potential in various classification and function approximation tasks, exploration of this model remains relatively limited compared to conventional deep learning architectures. Existing DNM variants generally still rely on end-to-end optimization via backpropagation, thus facing constraints in computational efficiency, sensitivity to initialization and activation function choice, and limited scalability on devices with restricted computational resources [29-31].

This study proposes the Randomized Dendritic Neuron Model (R-DNM) as a novel variant of DNM that preserves the biologically inspired neuromorphic structure, including the multiplicative gating mechanism in the dendritic layer, while significantly enhancing computational efficiency. The main innovation lies in the randomization and fixation of all parameters in the hidden layers (synapses, linear branches, and dendritic gating), which are randomly initialized from a standard Gaussian distribution and remain unoptimized during training [32-34]. Only the output weight matrix is trained analytically using a regularized least-squares solution (ridge regression) [35], [36]. This approach eliminates the need for iterative backpropagation on hidden layers, resulting in a much faster and memory-efficient training process, while providing an implicit ensemble effect that enhances robustness to data variations and numerical instability.

II. MATERIALS AND METHODS

This study follows a systematic methodological workflow as illustrated in Figure 1. The three benchmark medical datasets employed are the Pima Indians Diabetes Database (PIDD), Breast Cancer Wisconsin (Diagnostic), and Cardiovascular Disease Dataset, each exhibiting distinct clinical and epidemiological characteristics [37-39]. Prior to modeling, feature normalization was performed using StandardScaler, fitted exclusively on the training data per fold to prevent data leakage [40]. Evaluation was conducted via 10-fold cross-validation, with performance assessed through accuracy, precision, recall, F1-score, and log-loss to measure predictive capability and probabilistic calibration, while computational efficiency was quantified in seconds for training/inference time and in megabytes (MB) for total memory usage [41], [42]. The models compared include the conventional Dendritic Neuron Model (DNM), its multiclass extension Multi-Output Dendritic Neuron Model (MODN), the proposed Randomized Dendritic Neuron Model (R-DNM), as well as established baseline classifiers: Random Forest (RF), Support Vector Machine (SVM), and Multilayer Perceptron (MLP), enabling a comprehensive and fair assessment of both predictive performance and computational efficiency [43-46].

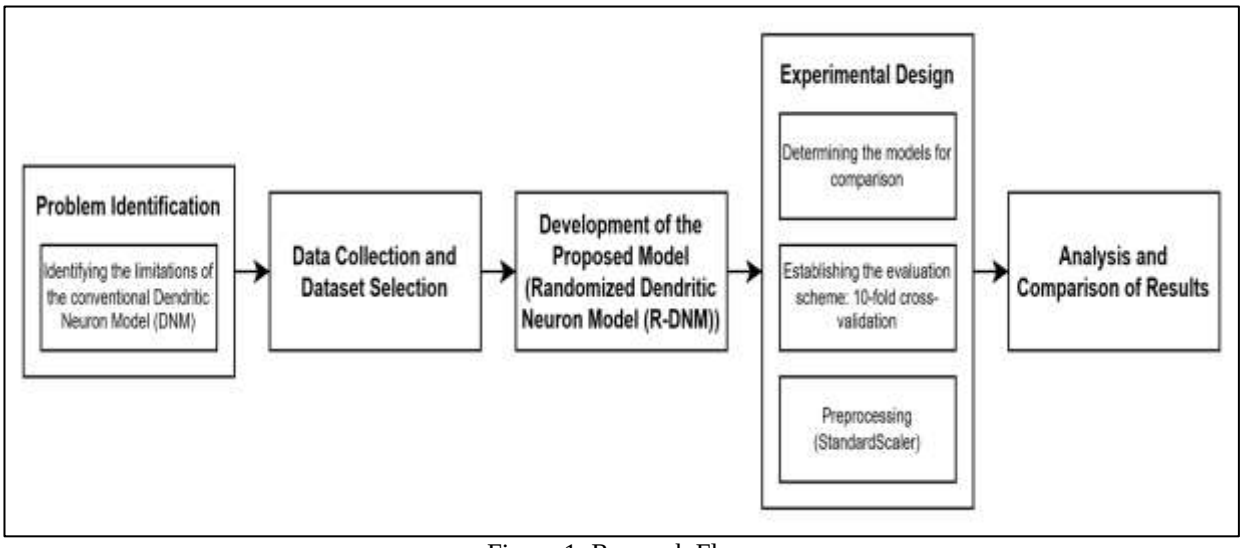

Figure 1: Research Flow.
Source: Authors, (2026).

II.1 DATASETS

This study evaluates the proposed Randomized Dendritic Neuron Model (R-DNM) using four publicly available medical datasets drawn from diverse clinical and epidemiological sources. The Pima Indians Diabetes Database (NIDDK) contains physiological measurements from adult Pima Indian women to assess diabetes risk factors. The Breast Cancer Wisconsin (Diagnostic) dataset provides quantitative nuclear features from fine needle aspiration biopsies for distinguishing benign from malignant tumors. The cardiovascular disease dataset integrates physiological and lifestyle risk factors from large-scale observational studies to support predictive modeling for prevention and early detection. Key characteristics, including class distributions and label interpretations, are summarized in Table 1, capturing realistic variability across healthcare domains.

Table 1: Summary of Datasets and Class Distribution.

Dataset	Class	Distribution	Proportion	Class Definition
Pima Indians Diabetes Database	0	500	0.6510	Non-diabetic
	1	268	0.3490	Diabetic
Breast Cancer Wisconsin (Diagnostic)	0	357	0.6274	Benign (Non-malignant tumor)
	1	212	0.3726	Malignant (Cancerous tumor)
Cardiovascular Disease Dataset	0	35,021	0.5003	No Cardiovascular Disease
	1	34,979	0.4997	Cardiovascular Disease

Source: Authors, (2026).

II.2 DENDRITIC NEURON MODEL (DNM)

Dendritic Neuron Model (DNM) is an artificial neuron model that is biologically inspired and designed to simulate information processing in biological neurons in a more realistic manner compared to conventional neuron models such as the multilayer perceptron [47], [48]. This model explicitly incorporates dendritic structures, in which non-linear operations occur not only at the soma but also along dendritic branches through a multiplicative gating mechanism. DNM consists of four main computational layers, namely the synapse layer, the linear branch layer, the dendritic gating layer, and the soma layer. The overall computational architecture of DNM is illustrated in Figure 2, which depicts the signal processing flow from input features to class probability outputs, with particular emphasis on multiplicative interactions at the dendritic layer.

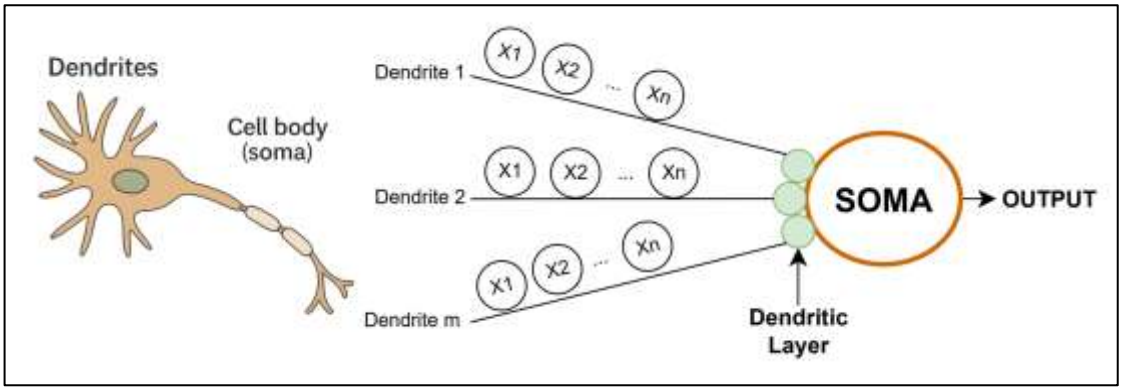

Figure 2: Architecture of the Dendritic Neuron Model.
Source: Authors, (2026).

The forward pass process in DNM is carried out sequentially for each input batch $X \in R^{b \times n_{features}}$, where b denotes the batch size, $n_{features}$ represents the number of features, and n_{hidden} indicates the number of hidden neurons. The following section provides a detailed explanation of each stage along with its complete mathematical formulation. Starting with the synapse layer, this layer simulates the aggregation of synaptic signals from presynaptic inputs. The initial linear operation is expressed in Formula (1):

$$S_1 = XW_{syn} + b_{syn} \tag{1}$$

Where $W_{syn} \in R^{n_{features} \times n_{hidden}}$ denotes the synaptic weight matrix, $b_{syn} \in R^{1 \times n_{hidden}}$ represents the synaptic bias vector that is replicated across b rows, and $S_1 \in R^{b \times n_{hidden}}$ corresponds to the initial synaptic output. Subsequently, in the linear branch layer, the synaptic output is further processed through a linear transformation to simulate signal propagation along dendritic branches. This operation is expressed in Formula (2):

$$S = S_1 W_{lin} + b_{lin} \tag{2}$$

Where $W_{lin} \in R^{n_{hidden} \times n_{hidden}}$ denotes the linear branch weight matrix, $b_{lin} \in R^{1 \times n_{hidden}}$ represents the linear bias vector, and $S \in R^{b \times n_{hidden}}$ corresponds to the final aggregated signal prior to the application of non-linearity. In a compact form, the synaptic–linear stage can be combined as expressed in Formula (3):

$$S = (XW_{syn} + b_{syn})W_{lin} + b_{lin} \tag{3}$$

Subsequently, in the non-linear activation layer, the signal S is passed through a non-linear activation function to produce the dendritic response. This operation is expressed in Formula (4):

$$H' = f_{input}(S) \tag{4}$$

Where $f_{input}(\cdot)$ denotes the activation function, for example sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$, hyperbolic tangent $tanh(x)$, or ReLU $max(0, x)$, and $H' \in R^{b \times n_{hidden}}$ represents the intermediate activation. In the dendritic gating layer, this layer simulates multiplicative modulation among dendritic branches. First, the gating signal is computed as shown in Formula (5):

$$Z = H'W_{dend} + b_{dend} \tag{5}$$

Where $W_{dend} \in R^{n_{hidden} \times n_{hidden}}$ denotes the dendritic weight matrix and $b_{dend} \in R^{1 \times n_{hidden}}$ represents the dendritic bias vector. Subsequently, the gating mechanism is applied as defined in Formula (6) and Formula (7):

$$D = f_{dendrite}(Z) \tag{6}$$

$$H = H' \odot D \tag{7}$$

Where $f_{dendrite}(\cdot)$ is the gating activation function, with sigmoid as the default choice to produce values in the range [0,1], $\odot$ denotes the Hadamard product, and $H \in R^{b \times n_{hidden}}$ corresponds to the final hidden representation after the gating process. In the soma layer, the hidden representation is mapped to the output space as expressed in Formula (8) and Formula (9):

$$Logits = H\beta \tag{8}$$

$$P = softmax(Logits)_j = \frac{exp(Logits_j)}{\sum_{k=1}^{n_{classes}} exp(Logits_k)}, \quad j = 1, \ldots, n_{classes} \tag{9}$$

Where $\beta \in R^{n_{hidden} \times n_{classes}}$ denotes the output weight matrix and $P \in R^{b \times n_{classes}}$ represents the class probability distribution. All DNM parameters are optimized end-to-end using backpropagation with the cross-entropy loss function and gradient-based optimizers such as Adam. The training and inference procedure of the Dendritic Neuron Model (DNM) is summarized at a high level in Figure 3.

***Input**: Training data, validation data, n_hidden, activation functions*
***Output**: Trained parameters and validation metrics*
Start *1. Randomly initialize all parameters (synaptic, linear, dendritic, output).* *2. Choose optimizer and Cross-Entropy loss.* *3. While not converged:* *- For each mini-batch:* *- Forward pass: synaptic → linear → activation → dendritic gating → output logits → softmax probabilities.* *- Compute loss.* *- Backpropagate and update all parameters.* *4. Inference on validation set: perform forward pass, obtain probabilities, evaluate metrics.* ***End***

Figure 3: Training and Inference of the Dendritic Neuron Model (DNM).
Source: Authors, (2026).

II.3 PROPOSED METHOD: RANDOMIZED DENDRITIC NEURON MODEL (R-DNM)

Randomized Dendritic Neuron Model (R-DNM) is a method proposed as a significant innovation over the conventional Dendritic Neuron Model (DNM). This approach fully preserves the neuromorphic architecture that is rich in biologically inspired non-linearities, including synaptic aggregation mechanisms, linear branch propagation, non-linear activation, and dendritic multiplicative gating. However, R-DNM introduces a strategy of randomizing the hidden-layer parameters to achieve substantially higher computational efficiency. Specifically, all parameters from the synaptic layer to the dendritic gating layer ($W_{syn}, b_{syn}, W_{lin}, b_{lin}, W_{dend}, b_{dena}$) are randomly initialized from a standard Gaussian distribution and remain fixed, that is, not optimized, throughout the entire training process. Only the output weight matrix β is trained analytically through a regularized closed-form least-squares solution, thereby eliminating the need for computationally and memory-intensive iterative backpropagation [49-52]. The R-DNM architecture is illustrated in Figure 4.

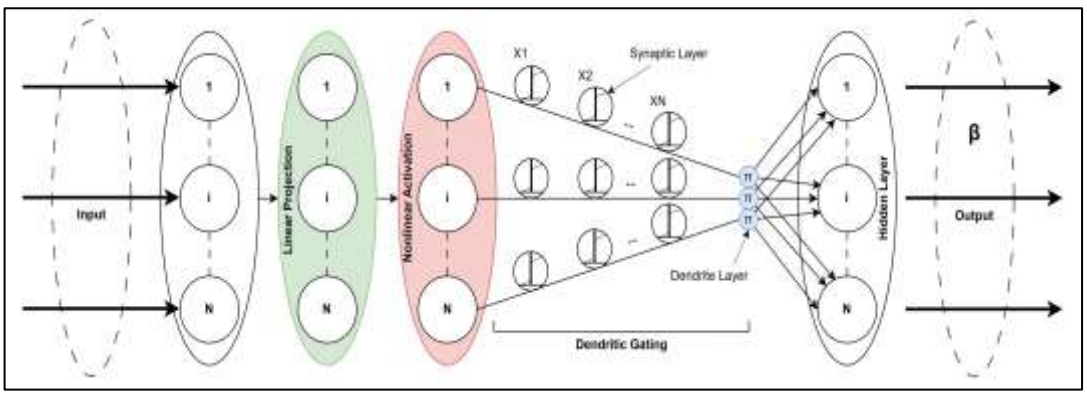

Figure 4: Architecture of the Proposed Method: Randomized Dendritic Neuron Model (R-DNM).
Source: Authors, (2026).

The forward pass process in R-DNM is identical to that of DNM up to the formation of the final hidden-layer representation H. Consequently, the computational stages directly follow the previously defined formulations, where the synapse and linear branch layers employ Formula (1)–Formula (3), the non-linear activation layer uses Formula (4), and the dendritic gating layer applies Formula (5)–Formula (7), such that the hidden representation $H \in R^{N \times n_{hidden}}$, for N data samples, is obtained through the same operations. The mapping to the output space also remains unchanged, namely the computation of the logits using Formula (8), followed by the calculation of the predictive probabilities using Formula (9), with numerical clamping applied. The fundamental difference lies in the optimization strategy. In R-DNM, since the hidden parameters are fixed, the hidden matrix H_{train} can be computed once through a forward pass without tracking gradients. The optimization process is therefore focused solely on β, by minimizing a regularized least-squares objective function (ridge regression), as defined in Formula (10):

$$\mathcal{J}(\beta) = \left|\left|H_{train}\beta - Y\right|\right|_F^2 + \lambda\left|\left|\beta\right|\right|_F^2 \tag{10}$$

Where $H_{train} \in R^{N_{train} \times n_{hidden}}$ denotes the hidden matrix obtained from the training data, $y_{train} \in \{1, \ldots, n_{classes}\}^{N_{train}}$ represents the target class labels for the training samples, $\lambda \geq 0$ is the ridge regularization hyperparameter, and $||\cdot||_F$ denotes the Frobenius norm, which is equivalent to $\sum_{i=1}^{N_{train}} \left|\left|(H_{train}\beta)_{i,:} - e_{y_i}\right|\right|_2^2 + \lambda \sum_{j,k} \beta_{j,k}^2$, where e_{y_i} denotes the canonical basis vector corresponding to the class label y_i. To obtain the optimal solution, the partial derivative of the objective function with respect to β is set to zero:

$$\frac{\partial \mathcal{J}}{\partial \beta} = 2H_{train}^{\top}(H_{train}\beta - Y) + 2\lambda\beta = 0 \tag{11}$$

Which yields

$$H_{train}^{\top}H_{train}\beta + \lambda\beta = H_{train}^{\top}Y \tag{12}$$

And can be rewritten as

$$(H_{train}^{\top}H_{train} + \lambda I)\beta = H_{train}^{\top}Y \tag{13}$$

Where $I \in R^{n_{hidden} \times n_{hidden}}$ is the identity matrix. The final closed-form solution is given in Formula (14):

$$\beta^* = (H_{train}^{\top}H_{train} + \lambda I)^{-1}H_{train}^{\top}Y \tag{14}$$

This approach allows the training of R-DNM to be completed in a single matrix computation step, with a computational complexity of $O(Nn_{hidden}^2 + n_{hidden}^3)$ due to the matrix inversion, which is substantially faster and more memory efficient than the iterative gradient-based optimization used in DNM. In addition, the randomization of hidden parameters provides an implicit ensemble effect that enhances robustness to noise and reduces overfitting. The complete training and inference procedure of the Randomized Dendritic Neuron Model (R-DNM) is summarized in Figure 5.

***Input**: Training data, validation data, n_hidden, λ (ridge), activation functions, random seed*
***Output**: Optimal output weights β* and validation metrics*
Start *1. Set random seed for reproducibility.* *2. Randomly initialize and fix all hidden-layer parameters (synaptic, linear branch, dendritic gating).* *3. Compute training hidden representation:* *- Perform full forward pass through synapse → linear branch → activation → dendritic gating.* *- Obtain hidden matrix H_train.* *4. Analytically solve ridge regression to compute optimal output weights β*.* *5. Inference on validation set:* *- Perform forward pass to obtain probabilities P_val.* *- Evaluate performance metrics against y_val.* ***End***

Figure 5: Training and Inference of the Randomized Dendritic Neuron Model (R-DNM).
Source: Authors, (2026).

II.4 EXPERIMENTAL SETUP AND EVALUATION

This study evaluates three neuromorphic models—the Dendritic Neuron Model (DNM), its multiclass extension Multi-Output Dendritic Neuron Model (MODN), and the proposed Randomized Dendritic Neuron Model (R-DNM)—alongside established baseline classifiers: Random Forest (RF), Support Vector Machine (SVM), and Multilayer Perceptron (MLP). All experiments were conducted under identical conditions across four medical datasets to ensure fair and reproducible comparisons.

MODN extends DNM directly to multiclass classification by expanding the output weight matrix dimensionality while preserving the internal neuromorphic processing pipeline. In contrast, R-DNM retains the same dendritic-based hidden architecture up to the hidden representation but introduces a key innovation: hidden-layer parameters are randomly initialized and fixed, with only the output weights learned analytically via closed-form ridge regression. This design substantially reduces computational complexity and memory usage compared to the iterative backpropagation required by DNM and MODN, enabling higher efficiency without compromising biologically inspired dendritic non-linearities [53], [54].

Model performance was assessed using accuracy, precision, recall, F1-score, and log-loss to evaluate predictive capability, with particular emphasis on precision–recall balance for imbalanced datasets and probabilistic calibration. Computational efficiency was measured through training and inference time as well as memory consumption. Features were normalized using StandardScaler fitted exclusively on training data in each fold to prevent data leakage. All experiments were executed on a Windows 11 Home 64-bit system equipped with an AMD Ryzen 5 6600H processor (~3.3 GHz, 12 cores), 16 GB RAM, and 8,290 MB total graphics memory (483 MB dedicated, 7,806 MB shared). The complete experimental workflow is detailed in Figure 6.

***Input**: Dataset X, labels y, k = 10 folds*
***Output**: Average classification and computational metrics*
Start
1. Load dataset and labels.
2. Partition data into 10 folds.
3. For each fold:
- Split into training and test sets.
- Normalize features using StandardScaler (fit on training only).
- Train model (DNM, MODN, R-DNM, RF, SVM, or MLP).
- Perform inference on test set.
- Compute accuracy, precision, recall, F1-score, log-loss, computation time, and memory usage.
4. Average all metrics across the 10 folds.
5. Report final results.
End

Figure 6: Experimental and Evaluation Procedure.
Source: Authors, (2026).

III. RESULTS AND DISCUSSIONS

The performance evaluation of the Dendritic Neuron Model (DNM), Multi-Output Dendritic Neuron Model (MODN), and the proposed Randomized Dendritic Neuron Model (R-DNM) on the Pima Indians Diabetes dataset demonstrates the clear superiority of R-DNM in terms of computational efficiency and overall effectiveness. As summarized in Table 2, R-DNM consistently requires only 0.0020–0.0024 seconds for training and inference with memory usage of 0.0009 MB across all activation function combinations, in stark contrast to DNM and MODN, which require 0.4301–0.6597 seconds and 0.0063–0.0080 MB, respectively. This substantial reduction in computational cost is primarily due to the fixed random initialization of hidden-layer parameters—including synaptic, linear branch, and dendritic gating layers—drawn from a standard Gaussian distribution, with only the output weights optimized analytically using regularized least-squares, thereby reducing training complexity to a single dominant matrix operation.

In terms of classification performance, R-DNM outperforms DNM and MODN in nearly all activation configurations, except for the tanh–tanh setting, which yields the poorest results for all models due to severe saturation and vanishing gradients during iterative backpropagation. The optimal configuration for R-DNM is relu–tanh, achieving the highest average accuracy of 0.7552 ± 0.0362, precision of 0.7546 ± 0.0359, recall of 0.7552 ± 0.0362, F1-score of 0.7529 ± 0.0343, and log-loss of 0.5668 ± 0.0218. Relative to DNM and MODN, R-DNM exhibits consistent accuracy gains across several configurations, including +0.0169 for sigmoid–sigmoid, +0.0247 for sigmoid–tanh, and +0.0338 for relu–tanh, accompanied by reduced standard deviations that indicate improved cross-fold stability. These improvements are attributed to the implicit ensemble effect of large-scale random hidden representations and the preserved effectiveness of dendritic multiplicative gating for selective nonlinear modulation, particularly when ReLU is employed at the input layer, resulting in lower numerical instability and substantially reduced log-loss compared to DNM and MODN.

Table 2: Performance Comparison of Dendritic Neuron Models with Different Activation Function Combinations on Pima Indians Diabetes Dataset.

Activation Function (Input - Dendrite)	Method	Accuracy	Precision	Recall	F1-score	Log-loss	Computation time (s)	Total memory (MB)
sigmoid - sigmoid	DNM	0.7123 ± 0.0490	0.7109 ± 0.0514	0.7123 ± 0.0490	0.7093 ± 0.0486	1.3550 ± 0.3666	0.6382 ± 0.0664	0.0077 ± 0.0011
	MODN	0.7123 ± 0.0490	0.7109 ± 0.0514	0.7123 ± 0.0490	0.7093 ± 0.0486	1.3550 ± 0.3666	0.4548 ± 0.0403	0.0080 ± 0.0031
	Proposed Method: R-DNM	0.7292 ± 0.0264	0.7300 ± 0.0266	0.7292 ± 0.0264	0.7268 ± 0.0246	0.5621 ± 0.0243	0.0022 ± 0.0011	0.0009 ± 0.0000
sigmoid - tanh	DNM	0.7110 ± 0.0471	0.7099 ± 0.0468	0.7110 ± 0.0471	0.7087 ± 0.0468	1.7905 ± 0.3539	0.6232 ± 0.0813	0.0073 ± 0.0006
	MODN	0.7110 ± 0.0471	0.7099 ± 0.0468	0.7110 ± 0.0471	0.7087 ± 0.0468	1.7905 ± 0.3539	0.4871 ± 0.0622	0.0074 ± 0.0010

	Proposed Method: R-DNM	0.7357 ± 0.0488	0.7389 ± 0.0434	0.7357 ± 0.0488	0.7348 ± 0.0461	0.5619 ± 0.0204	0.0021 ± 0.0009	0.0009 ± 0.0002
sigmoid - relu	DNM	0.7251 ± 0.0437	0.7233 ± 0.0409	0.7251 ± 0.0437	0.7191 ± 0.0426	3.2714 ± 0.5157	0.6316 ± 0.0735	0.0072 ± 0.0004
	MODN	0.7251 ± 0.0437	0.7233 ± 0.0409	0.7251 ± 0.0437	0.7191 ± 0.0426	3.2714 ± 0.5157	0.4459 ± 0.0268	0.0066 ± 0.0003
	Proposed Method: R-DNM	0.7331 ± 0.0411	0.7377 ± 0.0399	0.7331 ± 0.0411	0.7325 ± 0.0385	0.5724 ± 0.0239	0.0021 ± 0.0004	0.0009 ± 0.0000
tanh - sigmoid	DNM	0.7161 ± 0.0353	0.7169 ± 0.0333	0.7161 ± 0.0353	0.7147 ± 0.0343	1.7314 ± 0.3129	0.6283 ± 0.0776	0.0072 ± 0.0005
	MODN	0.7161 ± 0.0353	0.7169 ± 0.0333	0.7161 ± 0.0353	0.7147 ± 0.0343	1.7314 ± 0.3129	0.4528 ± 0.0208	0.0067 ± 0.0005
	Proposed Method: R-DNM	0.7356 ± 0.0453	0.7431 ± 0.0444	0.7356 ± 0.0453	0.7362 ± 0.0436	0.5731 ± 0.0276	0.0024 ± 0.0011	0.0009 ± 0.0002
tanh - tanh	DNM	0.6408 ± 0.0503	0.6314 ± 0.0503	0.6408 ± 0.0503	0.6332 ± 0.0484	2.6561 ± 0.3843	0.6597 ± 0.0869	0.0072 ± 0.0004
	MODN	0.6408 ± 0.0503	0.6314 ± 0.0503	0.6408 ± 0.0503	0.6332 ± 0.0484	2.6561 ± 0.3843	0.4623 ± 0.0512	0.0071 ± 0.0003
	Proposed Method: R-DNM	0.6432 ± 0.0458	0.6228 ± 0.0552	0.6432 ± 0.0458	0.6243 ± 0.0479	0.6399 ± 0.0153	0.0020 ± 0.0009	0.0009 ± 0.0000
tanh - relu	DNM	0.7056 ± 0.0400	0.6992 ± 0.0392	0.7056 ± 0.0400	0.6979 ± 0.0391	4.1604 ± 0.5175	0.6390 ± 0.0618	0.0072 ± 0.0007
	MODN	0.7056 ± 0.0400	0.6992 ± 0.0392	0.7056 ± 0.0400	0.6979 ± 0.0391	4.1604 ± 0.5175	0.4472 ± 0.0209	0.0071 ± 0.0005
	Proposed Method: R-DNM	0.7136 ± 0.0424	0.7227 ± 0.0488	0.7136 ± 0.0424	0.7143 ± 0.0433	0.5817 ± 0.0249	0.0020 ± 0.0009	0.0009 ± 0.0000
relu - sigmoid	DNM	0.7200 ± 0.0420	0.7160 ± 0.0494	0.7200 ± 0.0420	0.7126 ± 0.0425	4.1586 ± 0.5363	0.6541 ± 0.0806	0.0070 ± 0.0003
	MODN	0.7200 ± 0.0420	0.7160 ± 0.0494	0.7200 ± 0.0420	0.7126 ± 0.0425	4.1586 ± 0.5363	0.4607 ± 0.0450	0.0068 ± 0.0005
	Proposed Method: R-DNM	0.7435 ± 0.0525	0.7412 ± 0.0552	0.7435 ± 0.0525	0.7391 ± 0.0528	0.5679 ± 0.0238	0.0021 ± 0.0004	0.0009 ± 0.0001
relu - tanh	DNM	0.7214 ± 0.0484	0.7126 ± 0.0583	0.7214 ± 0.0484	0.7053 ± 0.0587	4.3003 ± 0.8294	0.6267 ± 0.0480	0.0072 ± 0.0003
	MODN	0.7214 ± 0.0484	0.7126 ± 0.0583	0.7214 ± 0.0484	0.7053 ± 0.0587	4.3003 ± 0.8294	0.4301 ± 0.0144	0.0070 ± 0.0005
	Proposed Method: R-DNM	0.7552 ± 0.0362	0.7546 ± 0.0359	0.7552 ± 0.0362	0.7529 ± 0.0343	0.5668 ± 0.0218	0.0022 ± 0.0006	0.0009 ± 0.0000
relu - relu	DNM	0.6901 ± 0.0285	0.6770 ± 0.0402	0.6901 ± 0.0285	0.6617 ± 0.0372	4.9407 ± 0.4547	0.6260 ± 0.0588	0.0072 ± 0.0004
	MODN	0.6901 ± 0.0285	0.6770 ± 0.0402	0.6901 ± 0.0285	0.6617 ± 0.0372	4.9407 ± 0.4547	0.4440 ± 0.0382	0.0069 ± 0.0004
	Proposed Method: R-DNM	0.7306 ± 0.0494	0.7250 ± 0.0530	0.7306 ± 0.0494	0.7240 ± 0.0512	0.5856 ± 0.0220	0.0023 ± 0.0012	0.0009 ± 0.0000

Source: Authors, (2026).

The evaluation of the Dendritic Neuron Model (DNM), Multi-Output Dendritic Neuron Model (MODN), and the proposed Randomized Dendritic Neuron Model (R-DNM) on the Breast Cancer Wisconsin (Diagnostic) dataset consistently demonstrates the computational advantage of R-DNM across all activation function combinations. As reported in Table 3, R-DNM achieves training and inference times ranging from 0.0023 to 0.0041 seconds with a memory usage of 0.0009 MB, whereas DNM and MODN require substantially longer execution times (0.4444–0.8046 seconds) and larger memory footprints (0.0063–0.0077 MB). This efficiency gain is a direct consequence of fixing randomly initialized hidden-layer parameters and optimizing only the output weights analytically via ridge regression, thereby avoiding iterative backpropagation and gradient storage.

In contrast, classification performance on this dataset generally favors DNM and MODN due to the extremely high feature separability and low intrinsic complexity of the task. The best overall results are achieved under the sigmoid–sigmoid configuration, where DNM and MODN obtain an accuracy and F1-score of 0.9719 ± 0.0195 with a log-loss of 0.1806 ± 0.1761, while R-DNM reaches an accuracy of 0.9473 ± 0.0283 and a higher log-loss of 0.3881 ± 0.0255. Similar patterns are observed for sigmoid–tanh (0.9596 versus 0.9508) and sigmoid–relu (0.9684 versus 0.9456) configurations. Notably, R-DNM outperforms both DNM and MODN in activation settings susceptible to numerical instability, such as relu–relu and relu–tanh, where gradient-based models experience dead ReLU and exploding gradients, while R-DNM maintains stable accuracy above 0.9290 and consistently low log-loss due to its gradient-independent formulation.

Table 3: Performance Comparison of Dendritic Neuron Models with Different Activation Function Combinations on Breast Cancer Wisconsin (Diagnostic) Dataset.

Activation Function (Input - Dendrite)	Method	Accuracy	Precision	Recall	F1-score	Log-loss	Computation time (s)	Total memory (MB)
sigmoid - sigmoid	DNM	0.9719 ± 0.0195	0.9738 ± 0.0180	0.9719 ± 0.0195	0.9719 ± 0.0196	0.1806 ± 0.1761	0.8046 ± 0.0895	0.0077 ± 0.0035
	MODN	0.9719 ± 0.0195	0.9738 ± 0.0180	0.9719 ± 0.0195	0.9719 ± 0.0196	0.1806 ± 0.1761	0.4444 ± 0.0159	0.0075 ± 0.0028
	Proposed Method: R-DNM	0.9473 ± 0.0283	0.9506 ± 0.0262	0.9473 ± 0.0283	0.9467 ± 0.0287	0.3881 ± 0.0255	0.0028 ± 0.0013	0.0009 ± 0.0000
sigmoid - tanh	DNM	0.9596 ± 0.0261	0.9629 ± 0.0225	0.9596 ± 0.0261	0.9595 ± 0.0261	0.3295 ± 0.2548	0.7351 ± 0.0624	0.0065 ± 0.0003
	MODN	0.9596 ± 0.0261	0.9629 ± 0.0225	0.9596 ± 0.0261	0.9595 ± 0.0261	0.3295 ± 0.2548	0.4418 ± 0.0096	0.0065 ± 0.0005
	Proposed Method: R-DNM	0.9508 ± 0.0291	0.9524 ± 0.0282	0.9508 ± 0.0291	0.9503 ± 0.0296	0.3815 ± 0.0224	0.0041 ± 0.0028	0.0009 ± 0.0000
sigmoid - relu	DNM	0.9684 ± 0.0270	0.9717 ± 0.0225	0.9684 ± 0.0270	0.9683 ± 0.0269	0.3294 ± 0.2895	0.7323 ± 0.0898	0.0067 ± 0.0003
	MODN	0.9684 ± 0.0270	0.9717 ± 0.0225	0.9684 ± 0.0270	0.9683 ± 0.0269	0.3294 ± 0.2895	0.4593 ± 0.0332	0.0066 ± 0.0004
	Proposed Method: R-DNM	0.9456 ± 0.0318	0.9477 ± 0.0306	0.9456 ± 0.0318	0.9451 ± 0.0324	0.4021 ± 0.0269	0.0033 ± 0.0022	0.0009 ± 0.0002
tanh - sigmoid	DNM	0.9438 ± 0.0463	0.9479 ± 0.0422	0.9438 ± 0.0463	0.9439 ± 0.0458	0.3774 ± 0.3364	0.7307 ± 0.0825	0.0067 ± 0.0004
	MODN	0.9438 ± 0.0463	0.9479 ± 0.0422	0.9438 ± 0.0463	0.9439 ± 0.0458	0.3774 ± 0.3364	0.4509 ± 0.0399	0.0065 ± 0.0004
	Proposed Method: R-DNM	0.9456 ± 0.0253	0.9486 ± 0.0238	0.9456 ± 0.0253	0.9451 ± 0.0258	0.3926 ± 0.0195	0.0032 ± 0.0013	0.0009 ± 0.0000
tanh - tanh	DNM	0.7647 ± 0.0511	0.7673 ± 0.0496	0.7647 ± 0.0511	0.7618 ± 0.0505	1.4982 ± 0.5172	0.7462 ± 0.1371	0.0069 ± 0.0007
	MODN	0.7647 ± 0.0511	0.7673 ± 0.0496	0.7647 ± 0.0511	0.7618 ± 0.0505	1.4982 ± 0.5172	0.4587 ± 0.0402	0.0066 ± 0.0003
	Proposed Method: R-DNM	0.6750 ± 0.0633	0.6745 ± 0.0630	0.6750 ± 0.0633	0.6740 ± 0.0630	0.6279 ± 0.0314	0.0112 ± 0.0247	0.0009 ± 0.0000
tanh - relu	DNM	0.9298 ± 0.0323	0.9326 ± 0.0320	0.9298 ± 0.0323	0.9292 ± 0.0323	0.9764 ± 0.5727	0.7097 ± 0.0670	0.0066 ± 0.0003
	MODN	0.9298 ± 0.0323	0.9326 ± 0.0320	0.9298 ± 0.0323	0.9292 ± 0.0323	0.9764 ± 0.5727	0.4582 ± 0.0605	0.0065 ± 0.0003
	Proposed Method: R-DNM	0.9280 ± 0.0298	0.9324 ± 0.0253	0.9280 ± 0.0298	0.9278 ± 0.0299	0.4216 ± 0.0241	0.0032 ± 0.0015	0.0009 ± 0.0000
relu - sigmoid	DNM	0.9508 ± 0.0280	0.9533 ± 0.0262	0.9508 ± 0.0280	0.9503 ± 0.0286	0.7846 ± 0.4474	0.7179 ± 0.0733	0.0067 ± 0.0004
	MODN	0.9508 ± 0.0280	0.9533 ± 0.0262	0.9508 ± 0.0280	0.9503 ± 0.0286	0.7846 ± 0.4474	0.4660 ± 0.0120	0.0065 ± 0.0003
	Proposed Method: R-DNM	0.9509 ± 0.0383	0.9512 ± 0.0385	0.9509 ± 0.0383	0.9506 ± 0.0387	0.4086 ± 0.0182	0.0030 ± 0.0011	0.0009 ± 0.0000
relu - tanh	DNM	0.9139 ± 0.0336	0.9180 ± 0.0311	0.9139 ± 0.0336	0.9125 ± 0.0358	1.3294 ± 0.4790	0.7290 ± 0.0813	0.0067 ± 0.0003
	MODN	0.9139 ± 0.0336	0.9180 ± 0.0311	0.9139 ± 0.0336	0.9125 ± 0.0358	1.3294 ± 0.4790	0.4480 ± 0.0177	0.0067 ± 0.0006
	Proposed Method: R-DNM	0.9298 ± 0.0368	0.9321 ± 0.0352	0.9298 ± 0.0368	0.9299 ± 0.0364	0.4259 ± 0.0234	0.0027 ± 0.0007	0.0009 ± 0.0000
relu - relu	DNM	0.5553 ± 0.0699	0.5577 ± 0.0663	0.5553 ± 0.0699	0.5535 ± 0.0665	7.0892 ± 1.1151	0.7069 ± 0.0455	0.0065 ± 0.0004
	MODN	0.5553 ± 0.0699	0.5577 ± 0.0663	0.5553 ± 0.0699	0.5535 ± 0.0665	7.0892 ± 1.1151	0.4483 ± 0.0522	0.0063 ± 0.0005
	Proposed Method: R-DNM	0.9333 ± 0.0406	0.9363 ± 0.0379	0.9333 ± 0.0406	0.9325 ± 0.0415	0.4638 ± 0.0195	0.0023 ± 0.0010	0.0009 ± 0.0000

Source: Authors, (2026).

The performance assessment of the Dendritic Neuron Model (DNM), Multi-Output Dendritic Neuron Model (MODN), and the proposed Randomized Dendritic Neuron Model (R-DNM) on the large-scale and balanced Cardiovascular Disease dataset (approximately 70,000 samples with a near 50:50 class ratio) demonstrates the sustained computational advantage of R-DNM under high data volume.

As summarized in Table 4, R-DNM requires 1.8955–2.0362 seconds for training and inference with memory usage in the range of 0.0015–0.0016 MB, whereas DNM and MODN demand substantially longer execution times (7.6023–9.0608 seconds) and higher memory consumption, reaching up to 2.2401 ± 6.6981 MB for DNM and 0.0068–0.0080 MB for MODN. This efficiency is achieved through fixed random initialization of hidden-layer parameters and single-step analytical optimization of output weights via ridge regression, thereby avoiding the linear growth of computational cost associated with iterative backpropagation as dataset size increases. In terms of classification performance, results differ from those observed on smaller datasets, with DNM and MODN outperforming R-DNM across most activation configurations. The best overall performance is obtained under the sigmoid–sigmoid setting, where DNM and MODN achieve an accuracy of 0.7288 ± 0.0046, an F1-score of 0.7285 ± 0.0046, and a log-loss of 1.0231 ± 0.0168, compared to R-DNM's accuracy of 0.6919 ± 0.0071.

Similar trends are observed for sigmoid–tanh and tanh–sigmoid combinations, reflecting the superior capacity of end-to-end gradient-based optimization to exploit the dataset's balanced distribution and moderate-to-high complexity. Nevertheless, R-DNM demonstrates clear advantages in activation settings prone to gradient instability, such as relu-based configurations, where DNM and MODN experience severe performance degradation (e.g., accuracy of 0.6333 and log-loss of 5.8437 for relu–relu), while R-DNM maintains stable accuracy of 0.7081 and a substantially lower log-loss of 0.6159. Overall, these findings emphasize the trade-off between predictive accuracy and computational efficiency, positioning R-DNM as a scalable and robust alternative for large-scale medical applications with constrained computational resources.

Table 4: Performance Comparison of Dendritic Neuron Models with Different Activation Function Combinations on Cardiovascular Disease Dataset.

Activation Function (Input - Dendrite)	Method	Accuracy	Precision	Recall	F1-score	Log-loss	Computation time (s)	Total memory (MB)
sigmoid - sigmoid	DNM	0.7288 ± 0.0046	0.7299 ± 0.0047	0.7288 ± 0.0046	0.7285 ± 0.0046	1.0231 ± 0.0168	8.3109 ± 0.8423	2.2401 ± 6.6981
	MODN	0.7288 ± 0.0046	0.7299 ± 0.0047	0.7288 ± 0.0046	0.7285 ± 0.0046	1.0231 ± 0.0168	7.6841 ± 0.1648	0.0080 ± 0.0030
	Proposed Method: R-DNM	0.6919 ± 0.0071	0.6920 ± 0.0070	0.6919 ± 0.0071	0.6918 ± 0.0071	0.6216 ± 0.0022	1.9929 ± 0.1804	1.7116 ± 5.1303
sigmoid - tanh	DNM	0.7253 ± 0.0048	0.7259 ± 0.0049	0.7253 ± 0.0048	0.7251 ± 0.0048	1.4522 ± 0.0583	8.8255 ± 0.8411	0.0073 ± 0.0004
	MODN	0.7253 ± 0.0048	0.7259 ± 0.0049	0.7253 ± 0.0048	0.7251 ± 0.0048	1.4522 ± 0.0583	7.8948 ± 0.1335	0.0069 ± 0.0005
	Proposed Method: R-DNM	0.6856 ± 0.0063	0.6857 ± 0.0063	0.6856 ± 0.0063	0.6855 ± 0.0063	0.6242 ± 0.0021	2.0078 ± 0.2596	0.0016 ± 0.0002
sigmoid - relu	DNM	0.7252 ± 0.0061	0.7257 ± 0.0061	0.7252 ± 0.0061	0.7251 ± 0.0061	2.8883 ± 0.1165	8.4216 ± 0.8724	0.0068 ± 0.0005
	MODN	0.7252 ± 0.0061	0.7257 ± 0.0061	0.7252 ± 0.0061	0.7251 ± 0.0061	2.8883 ± 0.1165	7.6023 ± 0.2015	0.0072 ± 0.0002
	Proposed Method: R-DNM	0.6926 ± 0.0052	0.6926 ± 0.0051	0.6926 ± 0.0052	0.6925 ± 0.0052	0.6214 ± 0.0021	2.0014 ± 0.2312	0.0016 ± 0.0003
tanh - sigmoid	DNM	0.7223 ± 0.0047	0.7226 ± 0.0047	0.7223 ± 0.0047	0.7222 ± 0.0047	1.6475 ± 0.0496	8.7765 ± 0.8171	0.0070 ± 0.0005
	MODN	0.7223 ± 0.0047	0.7226 ± 0.0047	0.7223 ± 0.0047	0.7222 ± 0.0047	1.6475 ± 0.0496	8.1751 ± 0.8727	0.0072 ± 0.0003
	Proposed Method: R-DNM	0.6888 ± 0.0047	0.6888 ± 0.0047	0.6888 ± 0.0047	0.6887 ± 0.0047	0.6223 ± 0.0021	2.0293 ± 0.2706	0.0016 ± 0.0002
tanh - tanh	DNM	0.6815 ± 0.0062	0.6816 ± 0.0062	0.6815 ± 0.0062	0.6815 ± 0.0062	2.3898 ± 0.0770	9.0608 ± 0.7901	0.0071 ± 0.0005
	MODN	0.6815 ± 0.0062	0.6816 ± 0.0062	0.6815 ± 0.0062	0.6815 ± 0.0062	2.3898 ± 0.0770	8.2727 ± 0.4531	0.0070 ± 0.0005
	Proposed Method: R-DNM	0.6396 ± 0.0065	0.6396 ± 0.0065	0.6396 ± 0.0065	0.6396 ± 0.0065	0.6477 ± 0.0024	2.0362 ± 0.2474	0.0015 ± 0.0002
tanh - relu	DNM	0.6598 ± 0.0062	0.6599 ± 0.0062	0.6598 ± 0.0062	0.6598 ± 0.0062	4.8822 ± 0.1136	8.6689 ± 0.8216	0.0070 ± 0.0004
	MODN	0.6598 ± 0.0062	0.6599 ± 0.0062	0.6598 ± 0.0062	0.6598 ± 0.0062	4.8822 ± 0.1136	8.0335 ± 0.4503	0.0068 ± 0.0006
	Proposed Method: R-DNM	0.6868 ± 0.0057	0.6868 ± 0.0057	0.6868 ± 0.0057	0.6867 ± 0.0057	0.6239 ± 0.0018	1.9384 ± 0.0393	0.0016 ± 0.0002
relu - sigmoid	DNM	0.6714 ± 0.0082	0.6716 ± 0.0082	0.6714 ± 0.0082	0.6713 ± 0.0082	4.8688 ± 0.1219	8.5826 ± 0.9172	0.0072 ± 0.0004
	MODN	0.6714 ± 0.0082	0.6716 ± 0.0082	0.6714 ± 0.0082	0.6713 ± 0.0082	4.8688 ± 0.1219	7.6836 ± 0.2463	0.0072 ± 0.0005
	Proposed Method: R-DNM	0.7045 ± 0.0060	0.7047 ± 0.0060	0.7045 ± 0.0060	0.7045 ± 0.0060	0.6161 ± 0.0023	1.8955 ± 0.0417	0.0016 ± 0.0003
relu - tanh	DNM	0.6528 ± 0.0084	0.6534 ± 0.0083	0.6528 ± 0.0084	0.6524 ± 0.0085	5.2602 ± 0.1470	8.4964 ± 0.8339	0.0073 ± 0.0004
	MODN	0.6528 ± 0.0084	0.6534 ± 0.0083	0.6528 ± 0.0084	0.6524 ± 0.0085	5.2602 ± 0.1470	7.8824 ± 0.1661	0.0071 ± 0.0002
	Proposed Method: R-DNM	0.7012 ± 0.0062	0.7014 ± 0.0062	0.7012 ± 0.0062	0.7011 ± 0.0062	0.6178 ± 0.0024	1.9599 ± 0.1057	0.0016 ± 0.0003
relu - relu	DNM	0.6333 ± 0.0061	0.6358 ± 0.0068	0.6333 ± 0.0061	0.6316 ± 0.0064	5.8437 ± 0.0952	8.2332 ± 0.7251	0.0073 ± 0.0003
	MODN	0.6333 ± 0.0061	0.6358 ± 0.0068	0.6333 ± 0.0061	0.6316 ± 0.0064	5.8437 ± 0.0952	7.5852 ± 0.1776	0.0070 ± 0.0004
	Proposed Method: R-DNM	0.7081 ± 0.0072	0.7083 ± 0.0072	0.7081 ± 0.0072	0.7080 ± 0.0072	0.6159 ± 0.0030	1.9344 ± 0.2140	0.0015 ± 0.0002

Source: Authors, (2026).

To provide a concise cross-dataset comparison, Figures 7 and 8 summarize the average F1-score and computation time, averaged across all activation combinations, for DNM, MODN, and R-DNM. As illustrated in Figure 7, R-DNM achieves the highest average F1-score on the Pima Indians Diabetes dataset, attaining 0.7205 ± 0.0144 compared to 0.6958 ± 0.0149 for both DNM and MODN, reflecting the robustness introduced by implicit ensemble effects arising from random hidden representations on small and moderately imbalanced data. In contrast, DNM and MODN exhibit stronger average performance on the highly separable Breast Cancer Wisconsin dataset (0.9113 ± 0.0377 versus 0.8834 ± 0.0394 for R-DNM) and on the balanced Cardiovascular Disease dataset, where average F1-scores converge around 0.6886–0.6887, highlighting clear dataset-dependent performance trade-offs among the models.

Figure 8 further emphasizes the computational efficiency of R-DNM across all datasets. R-DNM consistently exhibits substantially lower average runtimes, ranging from 0.0020 to 0.0040 seconds on the smaller datasets and approximately 1.9800 seconds on the Cardiovascular Disease dataset. These observations are supported by paired t-tests: on the Pima Indians Diabetes dataset, R-DNM

demonstrates statistically significant improvements over DNM and MODN in F1-score ($t = -3.7695$, $p = 0.0014$, Cohen's $d = 1.6860$) and computation time ($t = 84.2454–110.6988$, $p < 0.0001$, $d = 37.6760–49.5060$).

On the Breast Cancer Wisconsin dataset, differences in F1-score are not statistically significant ($p = 0.1231$, $d = 0.7240$), whereas computation time differences remain highly significant ($p < 0.0001$, $d = 12.1830–17.3060$). Overall, these results confirm that R-DNM offers a favorable balance between predictive performance and computational efficiency, particularly for small-to-medium datasets and resource-constrained medical applications.

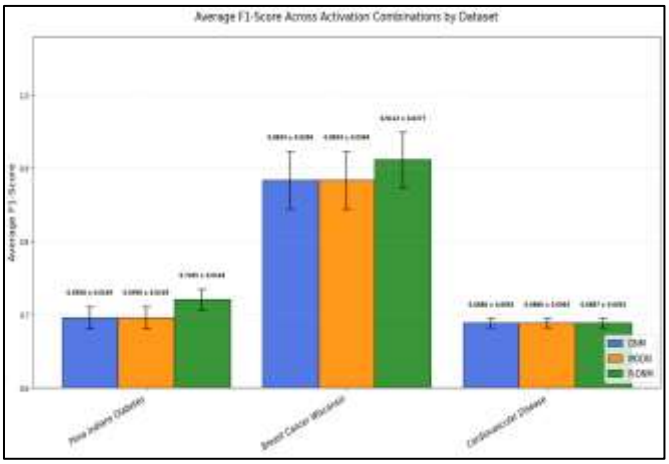

Figure 7: Average F1-Score Across Activation Combinations by Dataset.
Source: Authors, (2026).

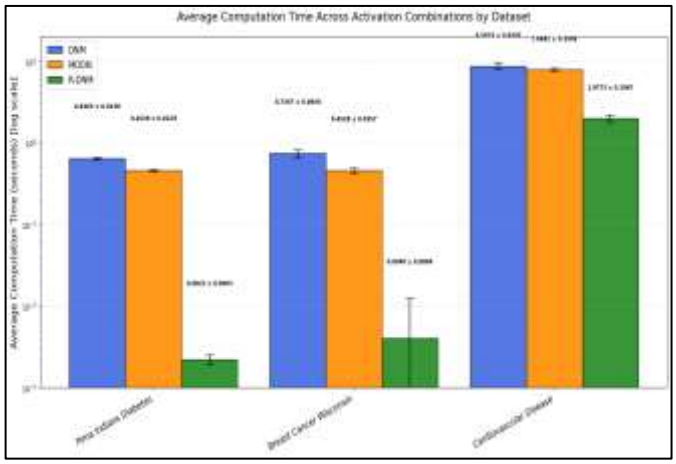

Figure 8: Average Computation Time Across Activation Combinations by Dataset.
Source: Authors, (2026).

The comparative evaluation of the best R-DNM configuration (ReLU–tanh activation) against established baseline classifiers—Random Forest (RF), Support Vector Machine (SVM), and Multilayer Perceptron (MLP)—on four medical datasets, as summarized in Table 5, indicates that R-DNM delivers competitive predictive performance while offering notable computational advantages. On the Pima Indians Diabetes dataset, R-DNM achieves an accuracy of 0.7552 ± 0.0362 and an F1-score of 0.7529 ± 0.0343, results that closely approach those of RF (0.7694 ± 0.0453) and SVM (0.7617 ± 0.0296) and are comparable to MLP (0.7525 ± 0.0435).

Similarly, on the Breast Cancer Wisconsin dataset, R-DNM attains an accuracy of 0.9333 ± 0.0406, slightly below the stronger baselines such as SVM and MLP (0.9754), while on the large-scale Cardiovascular Disease dataset it records an accuracy of 0.7081 ± 0.0072, marginally lower than RF (0.7251), SVM (0.7276), and MLP (0.7341). Despite these modest performance gaps, R-DNM consistently demonstrates favorable log-loss values in several cases and, more importantly, exhibits markedly superior efficiency, with training times ranging from 0.0022 to 1.9344 seconds and memory usage of approximately 0.0009–0.0015 MB, which is orders of magnitude lower than those required by RF (4.0143–39.6902 s), SVM (0.0548–1512.9478 s), and MLP (1.4114–43.9314 s).

Table 5: Comparison of Best R-DNM Configuration with Baseline Classifiers on Medical Datasets.

Dataset	Method	Accuracy	Precision	Recall	F1-score	Log-loss	Computation time (s)	Total memory (MB)
Pima Indians Diabetes	R-DNM (relu - tanh)	0.7552 ± 0.0362	0.7546 ± 0.0359	0.7552 ± 0.0362	0.7529 ± 0.0343	0.5668 ± 0.0218	0.0022 ± 0.0006	0.0009 ± 0.0000
	Random Forest	0.7694 ± 0.0453	0.7664 ± 0.0483	0.7694 ± 0.0453	0.7649 ± 0.0470	0.4857 ± 0.0573	4.0143 ± 0.4170	0.2591 ± 0.0046
	SVM	0.7617 ± 0.0296	0.7597 ± 0.0340	0.7617 ± 0.0296	0.7542 ± 0.0282	0.4936 ± 0.0455	0.1214 ± 0.0111	0.0368 ± 0.0008
	MLP	0.7525 ± 0.0435	0.7509 ± 0.0448	0.7525 ± 0.0435	0.7495 ± 0.0444	0.5760 ± 0.1116	5.0091 ± 0.3672	0.0593 ± 0.0013
Breast Cancer Wisconsin (Diagnostic)	R-DNM (relu - tanh)	0.9333 ± 0.0406	0.9363 ± 0.0379	0.9333 ± 0.0406	0.9325 ± 0.0415	0.4638 ± 0.0195	0.0023 ± 0.0010	0.0009 ± 0.0000
	Random Forest	0.9544 ± 0.0296	0.9565 ± 0.0279	0.9544 ± 0.0296	0.9541 ± 0.0301	0.1207 ± 0.0511	4.5808 ± 0.4896	0.2629 ± 0.0044
	SVM	0.9754 ± 0.0195	0.9767 ± 0.0183	0.9754 ± 0.0195	0.9754 ± 0.0195	0.0798 ± 0.0403	0.0548 ± 0.0079	0.0330 ± 0.0010
	MLP	0.9754 ± 0.0195	0.9766 ± 0.0179	0.9754 ± 0.0195	0.9752 ± 0.0199	0.0868 ± 0.0712	1.4114 ± 0.1809	0.0691 ± 0.0010
Cardiovascular Disease	R-DNM (relu - tanh)	0.7081 ± 0.0072	0.7083 ± 0.0072	0.7081 ± 0.0072	0.7080 ± 0.0072	0.6159 ± 0.0030	1.9344 ± 0.2140	0.0015 ± 0.0002
	Random Forest	0.7251 ± 0.0044	0.7256 ± 0.0043	0.7251 ± 0.0044	0.7249 ± 0.0044	0.5598 ± 0.0043	39.6902 ± 1.4447	0.2681 ± 0.0092
	SVM	0.7276 ± 0.0052	0.7285 ± 0.0053	0.7276 ± 0.0052	0.7273 ± 0.0052	0.5656 ± 0.0041	1512.9478 ± 251.9854	4.8254 ± 0.0067

	MLP	0.7341 ± 0.0046	0.7349 ± 0.0044	0.7341 ± 0.0046	0.7339 ± 0.0047	0.5450 ± 0.0042	43.9314 ± 6.8151	0.0354 ± 0.0017

Source: Authors, (2026).

To further contextualize the performance of the proposed Randomized Dendritic Neuron Model (R-DNM), Table 6 compares it with recent state-of-the-art methods evaluated on the same benchmark datasets. On the Pima Indians Diabetes Database (PIDD), R-DNM with the optimal configuration (ReLU activation at the input layer and tanh at the dendritic gating layer) achieves an accuracy of 0.7552 and an F1-score of 0.7529, which are competitive with ensemble-based approaches reported by (accuracy 0.7710, F1-score 0.6900) [16] and (accuracy 0.7700, F1-score 0.7100) [55]. Importantly, R-DNM exhibits significantly higher computational efficiency, requiring only about 0.0022 seconds for training and inference, whereas ensemble methods incur substantially higher costs due to iterative optimizatio and meta-learner training. This efficiency advantage makes R-DNM particularly suitable for resource-constrained, real-time applications on small to moderately imbalanced datasets such as PIDD. On the Breast Cancer Wisconsin (Diagnostic) dataset, R-DNM attains an accuracy of 0.9333 and an F1-score of 0.9325, slightly below the Intelligent Dendritic Neural Model optimized with Biogeography-Based Optimization (BBO) by (accuracy 0.9594) [29], a difference attributable to the dataset's high feature separability where evolutionary optimization yields marginal gains.

Nevertheless, R-DNM completes training and inference in approximately 0.0023 seconds, far faster than optimization-based approaches. Similarly, on the large-scale and balanced Cardiovascular Disease dataset, R-DNM achieves an accuracy of 0.7081 and an F1-score of 0.7080, approaching the performance of the attention-based Feature Decomposition Deep Learning (FDDL) model by (accuracy 0.7552, F1-score 0.7522) [18], while requiring only about 1.9344 seconds for training and inference. Overall, Table 6 highlights that R-DNM offers a strong performance–efficiency trade-off, delivering competitive predictive results with orders-of-magnitude improvements in computational speed and memory efficiency, making it a promising solution for real-world medical applications in resource-limited clinical settings.

Table 6: Comparison of the Proposed R-DNM (Optimal Configuration: ReLU–tanh) with Recent State-of-the-Art Methods on Benchmark Medical Datasets.

Dataset	Author(s)	Method(s)	Accuracy	F1-score
Pima Indians Diabetes	[16]	Ensemble learning (Decision Tree, Random Forest, Support Vector Machine, Neural Network, Logistic Regression)	0.7710	0.6900
	[55]	Ensemble models (Random Forest, Gradient Boosting, Multilayer Perceptron, XGBoost)	0.7700	0.7100
	This Study	Randomized Dendritic Neuron Model (ReLU activation at input layer – Tanh activation at dendritic layer)	0.7552	0.7529
Breast Cancer Wisconsin (Diagnostic)	[29]	Optimized Dendritic Neuron Model using Biogeography-Based Optimization	0.9594	-
	This Study	Randomized Dendritic Neuron Model (ReLU activation at input layer – Tanh activation at dendritic layer)	0.9333	0.9325
Cardiovascular Disease	[18]	Feature Decomposition-Based Deep Learning (FDDL) with attention mechanism and SHAP	0.7552	0.7522
	This Study	Randomized Dendritic Neuron Model (ReLU activation at input layer – Tanh activation at dendritic layer)	0.7081	0.7080

Source: Authors, (2026).

These results reveal a clear performance-efficiency trade-off modulated by dataset characteristics. On smaller or moderately imbalanced datasets such as PIDD, R-DNM delivers predictive accuracy and F1-scores competitive with both conventional baselines (RF, SVM, MLP) and recent state-of-the-art ensemble methods, benefiting from the implicit ensemble effect of randomized hidden representations and the robustness of dendritic gating. On datasets with high feature separability (Breast Cancer Wisconsin) or large-scale balanced data (Cardiovascular Disease), iterative optimization in end-to-end models and advanced deep architectures tends to yield marginally superior discrimination; however, R-DNM consistently offers orders-of-magnitude improvements in computational speed and memory usage, while demonstrating greater resilience to gradient instabilities (e.g., in ReLU-based configurations).

This positions R-DNM as a highly practical alternative for real-world deployment in resource-constrained clinical settings. The proposed Randomized Dendritic Neuron Model (R-DNM) preserves the biologically inspired non-linear multiplicative gating of dendritic structures while achieving extreme computational efficiency through fixed random hidden parameters and closed-form output optimization. Compared to both conventional classifiers and state-of-the-art approaches reported in recent literature, R-DNM provides competitive predictive performance across diverse medical tasks, coupled with training and inference speeds that are substantially faster—often by two to three orders of magnitude—making it particularly suitable for resource-limited environments.

Despite these advances, limitations remain, including slightly reduced accuracy on datasets with exceptionally high separability or large-scale balanced distributions, as well as limited exploration of activation functions and ridge regularization hyperparameters. Future work should explore hybrid training strategies that combine randomization with selective fine-tuning of hidden parameters, integration with multi-modal clinical data, external validation on real-world cohorts, and potential hardware implementation on neuromorphic platforms to further improve scalability and energy efficiency in digital health applications.

IV. CONCLUSIONS

This study introduces the Randomized Dendritic Neuron Model (R-DNM), a novel variant of the Dendritic Neuron Model that preserves the biologically inspired neuromorphic architecture, including the multiplicative gating mechanism in the dendritic layer, while achieving substantial improvements in computational efficiency through randomization and fixation of hidden-layer parameters combined with closed-form analytical training of the output weights via ridge regression. The proposed model delivers competitive predictive performance across benchmark medical datasets—Pima Indians Diabetes Database, Breast Cancer Wisconsin (Diagnostic), and

Cardiovascular Disease—often approaching or matching recent state-of-the-art ensemble, optimized dendritic, and attention-enhanced deep learning methods, while providing training and inference speeds that are orders of magnitude faster and memory requirements that are significantly lower.

These characteristics position R-DNM as a practical and scalable solution for early detection of non-communicable diseases in resource-constrained clinical environments. Nonetheless, limitations include modestly reduced accuracy on datasets exhibiting exceptionally high feature separability or large-scale balanced distributions, as well as limited exploration of activation function combinations and regularization hyperparameters. Future research directions should include the development of hybrid training strategies that integrate randomization with selective fine-tuning, extension to multi-modal clinical data, rigorous external validation using real-world patient cohorts, and hardware implementation on neuromorphic platforms to further advance scalability and energy efficiency in digital health applications.

V. AUTHOR'S CONTRIBUTION

Conceptualization: Michael Angello Qadosy Riyadi and Adinda Mariasti Dewi.
Methodology: Michael Angello Qadosy Riyadi.
Investigation: Michael Angello Qadosy Riyadi and Adinda Mariasti Dewi.
Discussion of results: Michael Angello Qadosy Riyadi and Adinda Mariasti Dewi.
Writing – Original Draft: Adinda Mariasti Dewi.
Writing – Review and Editing: Michael Angello Qadosy Riyadi and Adinda Mariasti Dewi.
Resources: Michael Angello Qadosy Riyadi and Adinda Mariasti Dewi.
Supervision: Michael Angello Qadosy Riyadi and Adinda Mariasti Dewi.
Approval of the final text: Michael Angello Qadosy Riyadi and Adinda Mariasti Dewi.

VI. REFERENCES

[1] R. Raj, R. Garg, and V. Kaur, "Data-Driven Insights into Cardiovascular Disease Mortality: A Global Analysis of Trends and Risks," in Proceedings of the Second International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS), Coimbatore, India, 2025, pp. 516–522. doi: 10.1109/ICC-ROBINS64345.2025.11086182.

[2] P. Saeedi et al., "Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation Diabetes Atlas, 9th edition," Diabetes Research and Clinical Practice, vol. 157, p. 107843, 2019, doi: 10.1016/j.diabres.2019.107843.

[3] G. A. Roth et al., "Global Burden of Cardiovascular Diseases and Risk Factors, 1990–2019: Update From the GBD 2019 Study," Journal of the American College of Cardiology, vol. 76, no. 25, pp. 2982–3021, 2020, doi: 10.1016/j.jacc.2020.11.010.

[4] A. N. Giaquinto et al., "Breast Cancer Statistics, 2022," CA: A Cancer Journal for Clinicians, vol. 72, no. 6, pp. 524–541, 2022, doi: 10.3322/caac.21754.

[5] Y. Zhang et al., "Global burden of female breast cancer: new estimates in 2022, temporal trend and future projections up to 2050 based on the latest release from GLOBOCAN," Journal of the National Cancer Center, vol. 5, no. 3, pp. 287–296, 2025, doi: 10.1016/j.jncc.2025.02.002.

[6] H. Sung et al., "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," CA: A Cancer Journal for Clinicians, vol. 71, no. 3, pp. 209–249, 2021, doi: 10.3322/caac.21660.

[7] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," Nature Medicine, vol. 25, no. 1, pp. 44–56, 2019, doi: 10.1038/s41591-018-0300-7.

[8] K. Khunti et al., "Diabetes and Multiple Long-term Conditions: A Review of Our Current Global Health Challenge," Diabetes Care, vol. 46, no. 12, pp. 2092–2101, 2023, doi: 10.2337/dci23-0035.

[9] K. Bhavsar, A. Abugabah, J. Singla, A. Ali, A. Bashir, and Nikita, "A Comprehensive Review on Medical Diagnosis Using Machine Learning," Computers, Materials & Continua, vol. 67, pp. 1997–2014, Dec. 2020, doi: 10.32604/cmc.2021.014943.

[10] A. Garg and V. Mago, "Role of machine learning in medical research: A survey," Computer Science Review, vol. 40, p. 100370, May 2021, doi: 10.1016/j.cosrev.2021.100370.

[11] S. Athisayamani, S. T, A. R. Singh, and others, "A novel double machine learning approach for detecting early breast cancer using advanced feature selection and dimensionality reduction techniques," Scientific Reports, vol. 15, p. 22971, 2025, doi: 10.1038/s41598-025-06426-7.

[12] M. Z. Awan, M. Zain, and K. Abodayeh, "Comparative analysis of machine learning models for breast cancer prediction and diagnosis: A dual-dataset approach," Indonesian Journal of Electrical Engineering and Computer Science, vol. Vol.34, pp. 2032–2044, Jun. 2024, doi: 10.11591/ijeecs.v34.i3.pp2032-2044.

[13] A. F. Ashour, M. M. Fouda, Z. M. Fadlullah, and M. I. Ibrahem, "Optimized Neural Networks for Diabetes Classification Using Pima Indians Diabetes Database," in 2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI), Mt Pleasant, MI, USA, 2024, pp. 1–7. doi: 10.1109/ICMI60790.2024.10585703.

[14] P. V. Sankar Ganesh and P. Sripriya, "A Comparative Review of Prediction Methods for Pima Indians Diabetes Dataset," in Computational Vision and Bio-Inspired Computing, S. Smys, J. M. R. S. Tavares, V. E. Balas, and A. M. Iliyasu, Eds., Cham: Springer International Publishing, 2020, pp. 735–750. doi: https://doi.org/10.1007/978-3-030-37218-7_83.

[15] T. J. H. Mim, Md. M. Hassan, B. K. Paul, and others, "Machine Learning Approaches for Cardiovascular Disease Prediction: A Comparative Study," Biomedical Materials & Devices, 2025, doi: 10.1007/s44174-025-00564-2.

[16] M. S. Reza, R. Amin, R. Yasmin, W. Kulsum, and S. Ruhi, "Improving diabetes disease patients classification using stacking ensemble method with PIMA and local healthcare data," Heliyon, vol. 10, no. 2, p. e24536, 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e24536.

[17] K. A. Ahmed et al., "Advancing breast cancer prediction: Comparative analysis of ML models and deep learning-based multi-model ensembles on original and synthetic datasets," PLOS ONE, vol. 20, no. 6, p. e0326221, 2025, doi: 10.1371/journal.pone.0326221.

[18] L. Yu et al., "Interpretable machine learning model for cardiovascular disease risk prediction: a feature decomposition-based study," BMC Public Health, vol. 25, no. 1, p. 3639, 2025, doi: 10.1186/s12889-025-24921-4.

[19] I. Araf, A. Idri, and I. Chairi, "Cost-sensitive learning for imbalanced medical data: a review," Artificial Intelligence Review, vol. 57, p. 80, 2024, doi: 10.1007/s10462-023-10652-8.

[20] S. A. Alowais et al., "Revolutionizing healthcare: the role of artificial intelligence in clinical practice," BMC Medical Education, vol. 23, no. 1, p. 689, 2023, doi: 10.1186/s12909-023-04698-z.

[21] X. Chen, H. Fan, W. Chen, Y. Zhang, D. Zhu, and S. Song, "Explaining a Logic Dendritic Neuron Model by Using the Morphology of Decision Trees," Electronics, vol. 13, no. 19, 2024, doi: 10.3390/electronics13193911.

[22] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic Neuron Model With Effective Learning Algorithms for Classification, Approximation, and Prediction," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 2, pp. 601–614, 2019, doi: 10.1109/TNNLS.2018.2846646.

[23] J. Ji, C. Tang, J. Zhao, T. Zheng, and Y. Todo, "A Survey on Dendritic Neuron Model: Mechanisms, Algorithms and Practical Applications," Neurocomputing, vol. 489, Mar. 2022, doi: 10.1016/j.neucom.2021.08.153.

[24] V. Takács et al., "Synaptic and dendritic architecture of different types of hippocampal somatostatin interneurons," PLOS Biology, vol. 22, no. 3, p. e3002539, 2024, doi: 10.1371/journal.pbio.3002539.

[25] Z. Wang, N. Yu, and F. Essaf, "A soma-synapses neuron model and FPGA implementation," Concurrency and Computation: Practice and Experience, vol. 35, no. 27, p. e7851, 2023, doi: 10.1002/cpe.7851.

[26] Y. Zhang, Y. Yang, X. Li, Z. Yuan, Y. Todo, and H. Yang, "A Dendritic Neuron Model Optimized by Meta-Heuristics with a Power-Law-Distributed Population Interaction Network for Financial Time-Series Forecasting," Mathematics, vol. 11, no. 5, p. 1251, 2023, doi: 10.3390/math11051251.

[27] Y. Yang, X. Li, H. Li, C. Zhang, Y. Todo, and H. Yang, "Yet Another Effective Dendritic Neuron Model Based on the Activity of Excitation and Inhibition," Mathematics, vol. 11, no. 7, p. 1701, 2023, doi: 10.3390/math11071701.

[28] C. Tang et al., "Visual Analytics of Learning Behavior Based on the Dendritic Neuron Model," in Knowledge Science, Engineering and Management, C. Cao, H. Chen, L. Zhao, J. Arshad, T. Asyhari, and Y. Wang, Eds., Singapore: Springer Nature Singapore, 2024, pp. 192–203. doi: https://doi.org/10.1007/978-981-97-5495-3_14.

[29] W. Xu, D. Jia, Z. Zhong, C. Li, and Z. Xu, "Intelligent Dendritic Neural Model for Classification Problems," Symmetry, vol. 14, no. 1, p. 11, 2022, doi: 10.3390/sym14010011.

[30] X. Luo and others, "Pruning method for dendritic neuron model based on dendrite layer significance constraints," CAAI Transactions on Intelligence Technology, vol. 8, no. 2, pp. 308–318, 2023, doi: 10.1049/cit2.12234.

[31] D. Jia, W. Xu, D. Liu, Z. Xu, Z. Zhong, and X. Ban, "Verification of Classification Model and Dendritic Neuron Model Based on Machine Learning," Discrete Dynamics in Nature and Society, vol. 2022, no. 1, p. 3259222, 2022, doi: https://doi.org/10.1155/2022/3259222.

[32] Y. M. Thant, S. Manzhos, M. Ihara, and M. Nukunudompanich, "On the Sufficiency of a Single Hidden Layer in Feed-Forward Neural Networks Used for Machine Learning of Materials Properties," Physchem, vol. 5, no. 1, p. 4, 2025, doi: 10.3390/physchem5010004.

[33] C. Peralez-González, J. Pérez-Rodríguez, and A. M. Durán-Rosal, "Boosting ridge for the extreme learning machine globally optimised for classification and regression problems," Scientific Reports, vol. 13, p. 11809, 2023, doi: 10.1038/s41598-023-38948-3.

[34] M. Toğaçar, B. Ergen, and Z. Cömert, "Application of breast cancer diagnosis based on a combination of convolutional neural networks, ridge regression and linear discriminant analysis using invasive breast cancer images processed with autoencoders," Medical Hypotheses, vol. 135, p. 109503, 2020, doi: 10.1016/j.mehy.2019.109503.

[35] J. Wang, F. Xie, F. Nie, and X. Li, "Generalized and Robust Least Squares Regression," IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 5, pp. 7006–7020, May 2024, doi: 10.1109/TNNLS.2022.3213594.

[36] I. Perfilieva, N. Madrid, M. Ojeda-Aciego, P. Artiemjew, and A. Niemczynowicz, "A critical analysis of the theoretical framework of the Extreme Learning Machine," Neurocomputing, vol. 621, p. 129298, 2025, doi: https://doi.org/10.1016/j.neucom.2024.129298.

[37] D. Chellappan and H. Rajaguru, "Generalizability of machine learning models for diabetes detection a study with nordic islet transplant and PIMA datasets," Scientific Reports, vol. 15, p. 4479, 2025, doi: 10.1038/s41598-025-87471-0.

[38] A. I. Jony and A. K. B. Arnob, "Deep Learning Paradigms for Breast Cancer Diagnosis: A Comparative Study on Wisconsin Diagnostic Dataset," Malaysian Journal of Science and Advanced Technology, vol. 4, no. 2, pp. 109–117, Mar. 2024, doi: https://doi.org/10.56532/mjsat.v4i2.245.

[39] M. Dorraki et al., "Improving Cardiovascular Disease Prediction With Machine Learning Using Mental Health Data: A Prospective UK Biobank Study," JACC. Advances, vol. 3, no. 9, p. 101180, 2024, doi: 10.1016/j.jacadv.2024.101180.

[40] X. Pei et al., "Robustness of machine learning to color, size change, normalization, and image enhancement on micrograph datasets with large sample differences," Materials & Design, vol. 232, p. 112086, 2023, doi: https://doi.org/10.1016/j.matdes.2023.112086.

[41] V. Lumumba, D. Sang, G. Njoka, D. Musyimi, and Kavita, "Comparative Analysis of Cross-Validation Techniques: LOOCV, K-folds Cross-Validation, and Repeated K-folds Cross-Validation in Machine Learning Models," American Journal of Theoretical and Applied Statistics, vol. 13, pp. 127–137, Oct. 2024, doi: 10.11648/j.ajtas.20241305.13.

[42] G. Naidu, T. Zuva, and E. M. Sibanda, "A Review of Evaluation Metrics in Machine Learning Algorithms," in Artificial Intelligence Application in Networks and Systems, R. Silhavy and P. Silhavy, Eds., Cham: Springer International Publishing, 2023, pp. 15–25. doi: https://doi.org/10.1007/978-3-031-35314-7_2.

[43] S. Paul, P. Ranjan, S. Kumar, and A. Kumar, "Disease Predictor Using Random Forest Classifier," in 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1–4. doi: 10.1109/ICONAT53423.2022.9726023.

[44] F. M. Megahed, Y. J. Chen, L. A. Jones-Farmer, and others, "Comparing classifier performance with baselines," Nature Methods, vol. 21, pp. 546–548, 2024, doi: 10.1038/s41592-024-02234-5.

[45] R. Guido, S. Ferrisi, D. Lofaro, and D. Conforti, "An Overview on the Advancements of Support Vector Machine Models in Healthcare Applications: A Review," Information, vol. 15, no. 4, p. 235, 2024, doi: 10.3390/info15040235.

[46] T. Bikku, "Multi-layered deep learning perceptron approach for health risk prediction," Journal of Big Data, vol. 7, p. 50, 2020, doi: 10.1186/s40537-020-00316-7.

[47] W. Xu et al., "A Dendritic Neuron Model for Disease Prediction," in 2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), 2021, pp. 1118–1122. doi: 10.1109/AEECA52519.2021.9574192.

[48] C. Tang, J. Ji, Y. Todo, A. Shimada, W. Ding, and A. Hirata, "Dendritic Neural Network: A Novel Extension of Dendritic Neuron Model," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 8, no. 3, pp. 2228–2239, 2024, doi: 10.1109/TETCI.2024.3367819.

[49] R. Giryes, G. Sapiro, and A. M. Bronstein, "Corrections to 'Deep Neural Networks With Random Gaussian Weights: A Universal Classification Strategy?,'" IEEE Transactions on Signal Processing, vol. 68, pp. 529–531, 2020, doi: 10.1109/TSP.2019.2961228.

[50] T. Uelwer, T. Hoffmann, and S. Harmeling, "Non-iterative Phase Retrieval with Cascaded Neural Networks," in Artificial Neural Networks and Machine Learning – ICANN 2021, vol. 12892, I. Farkaš, P. Masulli, S. Otte, and S. Wermter, Eds., in Lecture Notes in Computer Science, vol. 12892. , Cham: Springer, 2021. doi: 10.1007/978-3-030-86340-1_24.

[51] M. Uzair and N. Jamil, "Effects of Hidden Layers on the Efficiency of Neural Networks," in 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 2020, pp. 1–6. doi: 10.1109/INMIC50486.2020.9318195.

[52] D. Zhang, Z. Wang, and T. Masayoshi, "Neural-Network-Based Iterative Learning Control for Multiple Tasks," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 9, pp. 4178–4190, 2021, doi: 10.1109/TNNLS.2020.3017158.

[53] X. Wen, M. Zhou, A. Albeshri, L. Huang, X. Luo, and D. Ning, "Improving Classification Performance in Dendritic Neuron Models through Practical Initialization Strategies," Sensors, vol. 24, no. 6, p. 1729, 2024, doi: 10.3390/s24061729.

[54] Y. Ding, J. Yu, C. Gu, S. Gao, and C. Zhang, "A multi-in and multi-out dendritic neuron model and its optimization," Knowledge-Based Systems, vol. 286, p. 111442, 2024, doi: https://doi.org/10.1016/j.knosys.2024.111442.

[55] H. Zhou, S. Rahman, M. Angelova, C. R. Bruce, and C. Karmakar, "A robust and generalized framework in diabetes classification across heterogeneous environments," Computers in Biology and Medicine, vol. 186, p. 109720, 2025, doi: https://doi.org/10.1016/j.compbiomed.2025.109720.