applied sciences

MDPI

Article

Wart-Treatment Efficacy Prediction Using a CMA-ES-Based Dendritic Neuron Model

Shuangbao Song [1,*], Botao Zhang [1], Xingqian Chen [2], Qiang Xu [1] and Jia Qu [1]

[1] School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213164, China
[2] School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China
[*] Correspondence: leadingsong@cczu.edu.cn

Abstract: Warts are a prevalent condition worldwide, affecting approximately 10% of the global population. In this study, a machine learning method based on a dendritic neuron model is proposed for wart-treatment efficacy prediction. To prevent premature convergence and improve the interpretability of the model training process, an effective heuristic algorithm, i.e., the covariance matrix adaptation evolution strategy (CMA-ES), is incorporated as the training method of the dendritic neuron model. Two common datasets of wart-treatment efficacy, i.e., the cryotherapy dataset and the immunotherapy dataset, are used to verify the effectiveness of the proposed method. The proposed CMA-ES-based dendritic neuron model achieves promising results, with average classification accuracies of 0.9012 and 0.8654 on the two datasets, respectively. The experimental results indicate that the proposed method achieves better or more competitive prediction results than six common machine learning models. In addition, the trained dendritic neuron model can be simplified using a dendritic pruning mechanism. Finally, an effective wart-treatment efficacy prediction method based on a dendritic neuron model, which can provide decision support for physicians, is proposed in this paper.

Keywords: dendritic neuron model; CMA-ES; wart treatment; machine learning; computer-aided diagnosis

check for updates

Citation: Song, S.; Zhang, B.; Chen, X.; Xu, Q.; Qu, J. Wart-Treatment Efficacy Prediction Using a CMA-ES-Based Dendritic Neuron Model. *Appl. Sci.* **2023**, *13*, 6542. https://doi.org/10.3390/app13116542

Academic Editors: José Ignacio Abreu Salas, Yoan Gutiérrez Vázquez and Ansel Yoan Rodríguez González

Received: 2 April 2023
Revised: 16 May 2023
Accepted: 25 May 2023
Published: 27 May 2023

1. Introduction

With the application of artificial intelligence in many fields of society, such as industry [1], agriculture [2], bioinformatics [3,4], and biomedicine [5,6], the world has witnessed great developments as a result of artificial intelligence technology [7]. Especially in the medical industry, artificial intelligence technology has significantly improved healthcare and reduced costs [8–11]. Machine learning can combine medical data to generate appropriate predictive models. Excellent machine learning models can quickly and accurately predict diseases and assist doctors in making appropriate diagnoses for patients [12,13]. Machine learning models have become highly adaptable in the field of computer-aided diagnosis in recent years [14–16].

Warts are growths caused by human papillomavirus (HPV). There are many different types of warts that can result in different degrees of harm to the body [17–19]. HPV also has the potential to induce cancer when it infects specific areas of the body [20]. Due to the impact of warts on patients' lives, they usually need urgent treatment. Current clinical treatments for wart dermatosis include cryotherapy, immunotherapy, and destructive therapy. Different patients suffering from the same type of wart skin disease can have varying responses to the same treatment because of different symptoms and individual differences [21]. The cost of treatment and the pain experienced by the patient during the treatment process vary from one treatment method to another [22]. Therefore, choosing the right method can save patients money and reduce their pain during treatment. However, in clinical practice, physicians usually choose a treatment method for their patients using subjective judgment. In many cases, patients may require multiple treatments before achieving a cure.

The wart-treatment efficacy prediction problem, i.e., predicting whether a selected wart-treatment method is effective or not, remains a challenging task in computer-aided diagnosis. Machine learning methods can predict the appropriate treatment for wart patients, effectively eliminating their symptoms and avoiding repeated treatments. Khozeimeh et al. used a rule-based fuzzy logic system to predict the efficacy of different treatments for warts [23]. Akben et al. used an ID3 decision tree for wart-treatment efficacy prediction [24], where they converted the decision path generated by the decision tree into a fuzzy information graph. On the other hand, since data are key to machine learning-assisted medical diagnosis, Abdar et al. noticed that traditional machine learning models were less robust when performing wart-treatment efficacy prediction [25] because they could not effectively handle sample attributes with small values. To improve the accuracy of wart-treatment efficacy prediction, they proposed combining an adaptive particle swarm algorithm with an artificial immune recognition system to generate prediction models. The effect of data on prediction accuracy was similarly noted by Jha et al. [26]. They developed a fuzzy-rough-KNN algorithm based on efficient data feature generation and selection. In addition, the data imbalance problem is very common in currently available medically relevant datasets. Hu et al. used the Synthetic Minority Over-Sampling Technique (SMOTE) algorithm to balance raw data, addressing the data imbalance problem in wart-treatment efficacy prediction [27]. Although the above study improved the model from various perspectives to improve the accuracy of wart-treatment efficacy prediction, a machine learning model with higher accuracy and interpretability is still worthy of exploration by researchers.

Using the dendritic neuron model as a machine learning model has attracted significant attention in recent years. Ji et al. proposed using this model to address the classification problem [28] but noted that the performance of the model was limited due to the backpropagation algorithm easily falling into local convergence. To improve the classification performance of the model, Ji et al. used the states-of-matter search algorithm to improve the performance of the model [29]. Gao et al. also used a heuristic algorithm (a biogeography-based optimization algorithm) to train the dendritic neuron model [30]. Luo et al. used a decision-tree-based algorithm to initialize the weights of the dendritic neuron model [31], which effectively prevented the backpropagation algorithm from converging prematurely. The development of dendritic neuron models in several application areas has also attracted significant attention. Song et al. applied dendritic neuron models to wind-speed prediction and achieved excellent results [32]. He et al. improved the model structure based on the dendritic neuron model and applied the improved model to financial time-series prediction [33]. Tang et al. proposed the evolutionary dendritic neuron model, which has demonstrated good performance in the field of computer-aided diagnosis [34]. The performance of the dendritic neuron model has been greatly improved and successfully applied in several fields. However, to the best of the authors' knowledge, applying the dendritic neuron model to wart-treatment efficacy prediction has not yet been well explored. This motivates us to use the dendritic neuron model to address the wart-treatment efficacy prediction problem.

In this study, to further improve the performance of wart-treatment efficacy prediction, we used the covariance matrix adaptation evolution strategy (CMA-ES) to optimize the dendritic neuron model (DNM). The CMA-ES is considered more interpretable than other heuristic algorithms and has powerful optimization performance. The experimental results show that the improved DNM outperforms other comparable machine learning models in six metrics. It is worth mentioning that the specific pruning mechanism of the DNM can simplify the structure of the trained model. The proposed method can provide appropriate decision support for physicians. The contribution of this paper is threefold. First, a novel machine learning model, the DNM, is proposed for wart-treatment efficacy prediction. Second, the CMA-ES is incorporated as the training method of the DNM. Third, the experimental results demonstrate the advantages of the proposed CMA-ES-based dendritic neuron model in wart-treatment efficacy prediction.

The remainder of this paper is organized as follows. Section 2 presents a description of the DNM. Section 3 explains how the CMA-ES trains the DNM to address the wart-treatment efficacy prediction problem. Section 4 provides the experimental studies and discussion. Finally, the conclusions of this paper are presented in Section 5.

2. Materials

The proposed dendritic neuron model consists of four parts: the synaptic layer Y, the dendritic layer Z, the membrane layer V, and the cell body O. Its logical structure is shown in Figure 1.

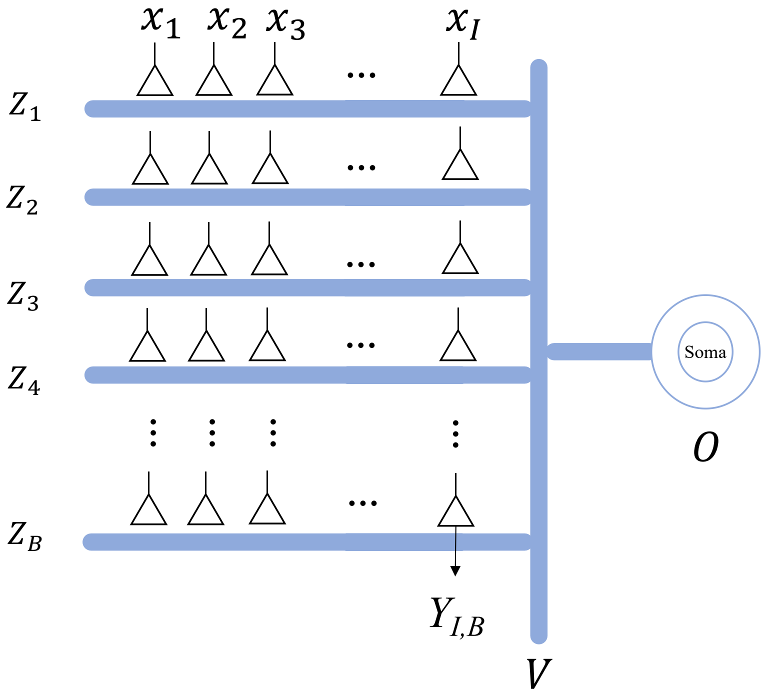


Figure 1. Logical structure of the proposed dendritic neuron model.

The synapse in the synaptic layer receives the input signal x_i and outputs $Y_{i,b}$ to the corresponding dendritic branch. The output $Y_{i,b}$ of the i-th ($i = 1, 2, \ldots, I$) synapse at the b-th ($b = 1, 2, \ldots, B$) dendritic branch can be expressed as follows:

$$Y_{i,b} = \frac{1}{1 + e^{-k(w_{i,b}x_i - q_{i,b})}} \tag{1}$$

where k is a predefined constant. $w_{i,b}$ and $q_{i,b}$ are the synaptic parameters to be optimized. Four different synaptic connection states can be identified according to the different $w_{i,b}$ and $q_{i,b}$ values, as shown in Figure 2. The different synaptic connection states affect the simplified pruning operation of the model. The determination of the different connection states can be found in the literature [35].

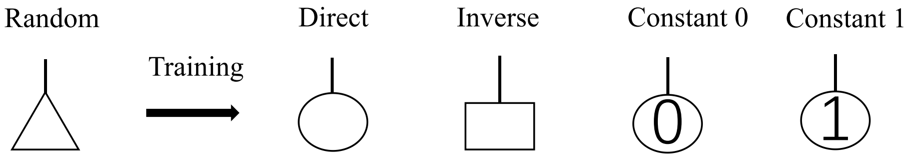


Figure 2. The random synaptic connection state is transformed into one of four synaptic connection states after training.

The dendritic layer receives signals from the synaptic layer and outputs Z_b to the membrane layer by performing a cumulative multiplication operation. The b-th dendritic branch can be expressed as follows:

$$Z_b = \prod_{i=1}^{I} Y_{i,b} \tag{2}$$

The membrane layer gathers the signals of all of the dendritic branches and transmits them to the cell body. The membrane layer can be represented by a large-scale summation operation, which is expressed as follows:

$$V = \sum_{b=1}^{B} Z_b \tag{3}$$

The cell body receives the output V of the membrane layer and transforms the signal V into the probability O using a sigmoid function, which is expressed as follows:

$$O = \frac{1}{1 + e^{-k(V-\gamma)}} \tag{4}$$

where γ is defined as the threshold of the cell body.

The pruning strategy of the dendritic neuron model is based on the effect of the synapses in the constant 0 connection state. The constant 0 connection causes the output value of the dendritic branch to be close to zero, according to Equation (2). Since this dendritic branch has a minimal effect on the calculation of the membrane layer according to Equation (3), this dendritic branch connected with a synapse in the constant 0 connection state can be pruned. An example of the specific dendritic pruning mechanism is shown in Figure 3. Figure 3a shows the trained DNM before pruning. Figure 3b shows the structure of the pruned DNM where the dendritic branches connected with the synapses in the constant 0 connection state are pruned.

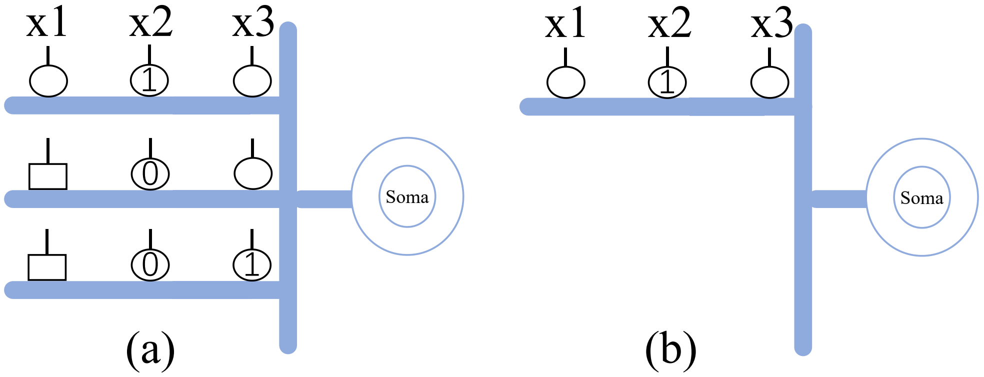


Figure 3. An example of the specific dendritic pruning mechanism for the proposed dendritic neuron model. The trained DNM is shown in (**a**), and the pruned DNM is shown in (**b**).

3. Methods

The training process of the dendritic neuron model is shown in Figure 4. First, the original data are normalized. Then, the synaptic parameters of the DNM are optimized using the CMA-ES. Finally, a DNM for wart-treatment efficacy prediction is obtained.

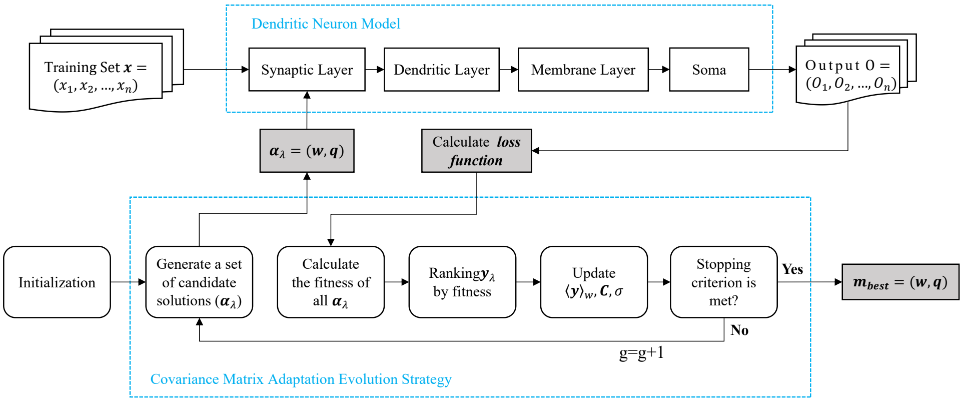

Dendritic Neuron Model
Training Set $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$
Synaptic Layer
Dendritic Layer
Membrane Layer
Soma
Output $O = (O_1, O_2, \ldots, O_n)$
$\boldsymbol{\alpha}_\lambda = (\boldsymbol{w}, \boldsymbol{q})$
Calculate *loss function*
Initialization
Generate a set of candidate solutions ($\boldsymbol{\alpha}_\lambda$)
Calculate the fitness of all $\boldsymbol{\alpha}_\lambda$
Ranking$\boldsymbol{y}_\lambda$ by fitness
Update $\langle \boldsymbol{y} \rangle_w, \boldsymbol{C}, \sigma$
Stopping criterion is met?
Yes
No
g=g+1
$\boldsymbol{m}_{best} = (\boldsymbol{w}, \boldsymbol{q})$
Covariance Matrix Adaptation Evolution Strategy

Figure 4. The flow chart of training the DNM using the CMA-ES.

3.1. Covariance Matrix Adaptation Evolution Strategy

The backpropagation algorithm is the most widely used optimization algorithm but it easily falls into local convergence in the process of DNM optimization. To prevent premature convergence of the model optimization process, Ji et al. attempted to optimize the dendritic neuron model using heuristic algorithms [29,30,35]. These heuristic algorithms can effectively help multivariate functions escape from local convergence. However, these heuristic algorithms are considered to lack interpretability. The CMA-ES is a powerful evolutionary algorithm and its optimization process can be interpreted as a form of natural gradient descent [36,37]. In actual optimization problems, multivariate functions are often very complex and it is difficult to obtain the corresponding Hessian matrix. The CMA-ES adjusts the covariance matrix of the multivariate function to approximate the Hessian matrix of the multivariate function [38,39]. The CMA-ES draws on the exploration and exploitation of the search strategy. Using the CMA-ES to optimize the DNM can effectively avoid local convergence. The key components of the CMA-ES are described below.

Optimization prerequisites: The Hessian matrix is positive definite, and the fitness function has a minimal value. The covariance matrix and Hessian matrix are inverse matrices of each other. The CMA-ES obtains the minimal value by updating the covariance matrix, and each update of the covariance matrix must satisfy the matrix positive definite, i.e., the matrix eigenvalues $\lambda_{max}/\lambda_{min} \geq 1$.

Candidate solution update: The CMA-ES has ω candidate solutions, where $\omega = 4 + \lfloor 3\ln n \rfloor$ and n is the dimensionality of the solution vector. Each candidate solution $\boldsymbol{\alpha}_\lambda$ is generated according to the corresponding multidimensional Gaussian distribution $\mathbf{y}_\lambda \sim N(\mathbf{m}, \mathbf{C})$. $\mathbf{m}$ is the mean vector of $\mathbf{y}_\lambda$. $\mathbf{C}$ is the covariance matrix of $\mathbf{y}_\lambda$, which determines the Gaussian distribution. Each candidate solution is updated, as shown in Equation (5). The calculations of $\mathbf{C}$ and σ are described later.

$$\boldsymbol{\alpha}_\lambda = \mathbf{m} + \sigma \mathbf{y}_\lambda \sim N(\mathbf{m}, \mathbf{C}) \tag{5}$$

Then, all of the candidate solutions are evaluated using the fitness function.

Overall Gaussian distribution update: To enable the fitness function to converge quickly, $\boldsymbol{\alpha}_\lambda$ corresponding to $\mathbf{y}_\lambda$ needs to be ranked, and those ranked after μ need to be filtered out, where $\mu = \omega/2$. A weight w_i' is assigned to the remaining excellent Gaussian distribution. The top μ weight w_i' is normalized to w_i. The overall Gaussian distribution $< \mathbf{y} >_w$ is updated based on the top μ excellent Gaussian distribution $\mathbf{y}_{i:w}$. The formula for calculating the weight w_i' is shown in Equation (6) and the formula for calculating $< \mathbf{y} >_w$ is shown in Equation (7).

$$w_i' = \frac{\ln(\lambda + 1)}{2} - \ln i \ , \ \ i = 1, 2, \ldots, \mu \tag{6}$$

$$< \mathbf{y} >_w = \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:w} \;\; , \;\; where \sum_{i=1}^{\mu} w_i = 1 \tag{7}$$

Covariance matrix update: The covariance matrix $\mathbf{C}$ update needs to combine the update history path of $\mathbf{C}$, which is also called the covariance matrix evolution path $\mathbf{P}_c$. The $\mathbf{P}_c$ update needs to combine $< \mathbf{y} >_w$. The evolution path $\mathbf{P}_c$ is updated using Equation (8). The covariance matrix $\mathbf{C}$ update requires a combined calculation of $\mathbf{P}_c$ and $\mathbf{y}_{i:w}$, which is calculated using Equation (9).

$$\mathbf{P}_c \leftarrow (1 - c_1)\mathbf{P}_c + h_\sigma \sqrt{c_c(2 - c_c)\mu_{eff}} < \mathbf{y} >_w \tag{8}$$

$$\mathbf{C} \leftarrow (1 + c_1 \delta(h_\sigma) - c_1 - c_\mu \sum w_j)\mathbf{C} + c_1 \mathbf{P}_c \mathbf{P}_c^T + c_\mu \sum_{i=1}^{\omega} w_i \mathbf{y}_{i:w} \mathbf{y}_{i:w}^T \tag{9}$$

The learning-rate parameters c_1, c_μ, and c_c are controlled by the parameter μ_{eff} [40]. h_σ is the Heaviside function, which takes different values according to σ and the number of iterations g. $\delta(h_\sigma)$ is the parameter that automatically selects the exploration search strategy or the exploitation search strategy by adjusting the covariance matrix. It is calculated as follows:

$$\delta(h_\sigma) = (1 - h_\sigma)c_c(2 - c_c) \tag{10}$$

Step-size update: The update strategy of the step size σ is similar to the $\mathbf{C}$ update strategy. It needs to combine the step-size evolutionary path $\mathbf{P}_\sigma$. The $\mathbf{P}_\sigma$ update requires a combination of $\mathbf{C}$ and $< \mathbf{y} >_w$. The step size σ is updated according to the ratio of $\mathbf{P}_\sigma$ to $\mathbf{P}_\sigma$ expectation $E\|N(\mathbf{0}, \mathbf{I})\|$, where $\mathbf{I}$ is the unit matrix. The step-size evolutionary path $\mathbf{P}_\sigma$ is updated using Equation (11) and the step size σ is updated using Equation (12).

$$\mathbf{P}_\sigma \leftarrow (1 - c_c)\mathbf{P}_\sigma + \sqrt{c_c(2 - c_\sigma)\mu_{eff}} \mathbf{C}^{-1/2} < \mathbf{y} >_w \tag{11}$$

$$\sigma \leftarrow \sigma exp(\frac{c_\sigma}{d_\sigma} \frac{\|\mathbf{P}_\sigma\|}{E\|N(\mathbf{0}, \mathbf{I})\|} - 1) \tag{12}$$

where the learning-rate parameters c_σ and d_σ are set according to the literature [40].

3.2. Applying the CMA-ES to Train the DNM

Figure 4 shows the training process of the DNM using the CMA-ES. The CMA-ES follows the general framework of evolutionary algorithms. It optimizes the problem by iteratively evolving a population of candidate solutions using the aforementioned operations. Since the DNM has two parameter vectors, $\mathbf{w}$ and $\mathbf{q}$, to be optimized, these two vectors form the solution vector of the CMA-ES algorithm as follows:

$$\boldsymbol{\alpha}_\lambda = \{x_\lambda^1, x_\lambda^2, \ldots, x_\lambda^{2 \cdot I \cdot B}\} = \{w_{1,1}, w_{1,2}, \ldots, w_{I,B}, q_{1,1}, q_{1,2}, \ldots, q_{I,B}\} \tag{13}$$

where $\boldsymbol{\alpha}_\lambda$ is the λ-th candidate solution in the population of the CMA-ES. The loss function of the DNM is calculated as the fitness value of $\boldsymbol{\alpha}_\lambda$. Finally, the CMA-ES terminates when the stopping criterion is met, and the optimal solution $\mathbf{m}_{\mathbf{best}}$ is outputted.

The mean square error (MSE) is commonly used as the loss function of the DNM. It can be calculated as follows:

$$MSE = \frac{1}{2S} \sum_{i=1}^{S} (O_i - T_i)^2 \tag{14}$$

where O_i is the actual output of the DNM, T_i is the label value, and S is the number of data samples.

When the dataset is unbalanced and the MSE function is used as the loss function, it makes the machine learning model more inclined to predict classes with large sample numbers [41]. Data sampling is commonly used in many studies to address data imbalance, but such methods can have an unexpected impact on the data: undersampling can result in missing data, oversampling is blind in generating the data, and data sampling, in general, can easily marginalize data [42–44]. In this study, we attempted to use a focal loss (FL) [45] to address the data imbalance problem. The FL function makes is designed to bring attention to the imbalanced data during the model training process, allowing the model to focus more attention on a fewer number of class samples in the dataset. It can improve the accuracy of the hard-to-classify samples by increasing the weights of fewer samples. For binary classification, the FL adds a power modifier $(1 - p_i)^u$ to the cross entropy and can be calculated as follows:

$$FL = -\frac{1}{S}\sum_{i=1}^{S}(1 - p_i)^u \log(p_i) \tag{15}$$

$$p_i = \begin{cases} O_i & \text{, if } T_i = 1 \\ 1 - O_i & \text{, otherwise} \end{cases} \tag{16}$$

where the output O_i of the DNM indicates the probability of the prediction with the label $T_i = 1$. u is a positive constant and is set to 2. A larger value of p_i indicates that the model's prediction is closer to the ground truth.

4. Experimental Studies

4.1. The Datasets of Wart-Treatment Efficacy

Two datasets collected from patients with wart skin disease were obtained from the Dermatology Department of Ghaem Hospital in Mashhad [46]. They can be accessed via the UCI Machine Learning Repository. The first dataset contained data from 90 patients treated with cryotherapy and each sample contained 6 features. The second dataset contained data from 90 patients treated with immunotherapy and each sample contained 7 features. The details of the cryotherapy dataset and the immunotherapy dataset are shown in Tables 1 and 2, respectively. There are 48 successful treatment cases and 42 unsuccessful treatment cases in the cryotherapy dataset. The immunotherapy dataset is subject to data imbalance, and there are 71 and 19 successful and unsuccessful treatment cases, respectively.

Table 1. The features of the cryotherapy dataset.

Feature No.	Feature Name	Value (Amount)
1	Gender	Man (47); Woman (43)
2	Age	15–67
3	Time elapsed before treatment (months)	0–12
4	Number of warts	1–12
5	Type of wart (count)	Common (54); Plantar (9); Both (27)
6	The surface area of warts (mm^2)	4–750

Table 2. The features of the immunotherapy dataset.

Feature No.	Feature Name	Value (Amount)
1	Gender	Man (41); Woman (49)
2	Age	15–56
3	Time elapsed before treatment (months)	0–12
4	Number of warts	1–19
5	Type of wart (count)	Common (47); Plantar (22); Both (21)
6	The surface area of warts (mm^2)	6–900
7	Induration diameter of initial test (mm)	5–70

4.2. Experimental Configuration

All algorithms in this study were implemented using Python 3.8. The experiments were executed on a Windows 10 computer with an AMD Ryzen7 3.59 GHz CPU. The comparison algorithm used in this study was based on the scikit-learn library [47]. The evaluation metrics included classification accuracy, sensitivity, specificity, precision, F1 score, and AUC value. The confusion matrix is shown in Table 3. The formulae for the above metrics are shown in Equations (17)–(21). Each dataset was randomly divided into a training set and a test set (70%∼30%). In addition, the DNM input signals (data features) of each dataset were normalized within the range of [0,1] before training.

Table 3. Confusion matrix.

Actual **Predicted**	**Treatment Success**	**Treatment Failure**
Treatment success	TP	FP
Treatment failure	FN	TN

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{17}$$

$$sensitivity = recall = \frac{TP}{TP + FN} \tag{18}$$

$$specificity = \frac{TN}{FP + TN} \tag{19}$$

$$precision = \frac{TP}{TP + FP} \tag{20}$$

$$F1score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{21}$$

4.3. Optimization Performance of CMA-ES

The DNM has three hyperparameters: the constant parameter k, the number of dendritic branches B, and the threshold γ. In the experiments, the standard $L_{16}(4^3)$ orthogonal array of Taguchi's method [48] was used to select the appropriate hyperparameters. Each hyperparameter has four different values, which are listed in Table 4. According to the orthogonal array, there were 16 parameter combinations. Each parameter combination was repeated 30 times to verify its stability, and in each experiment, the optimization algorithm underwent 100 iterations. The hyperparameters with the best classification results were selected based on the accuracy of the data obtained from the validation experiments. The results of the tuned hyperparameters are shown in Table 5.

Table 4. Values of the hyperparameters of the DNM.

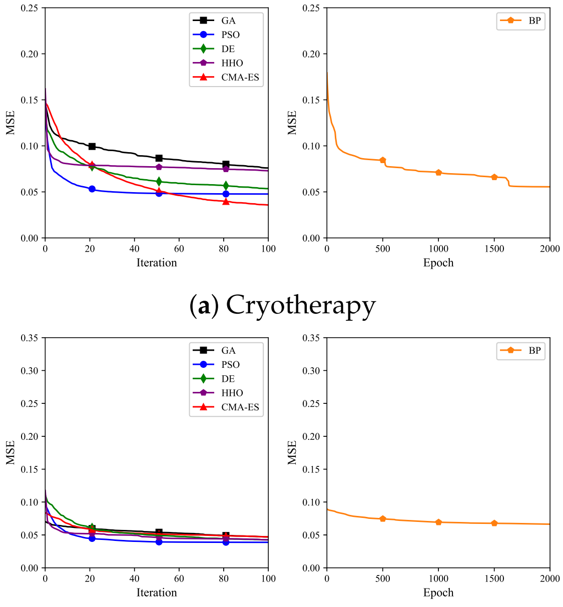
Hyperparameter	Value 1	Value 2	Value 3	Value 4
k	2	5	8	10
B	N_{fea} [a]	N_{fea} +2	N_{fea} +4	N_{fea} +6
γ	0.2	0.4	0.6	0.8

[a] the number of features.

Table 5. The hyperparameter settings of the DNM.

Dataset	k	B	γ
Cryotherapy	2	12	0.8
Immunotherapy	8	13	0.4

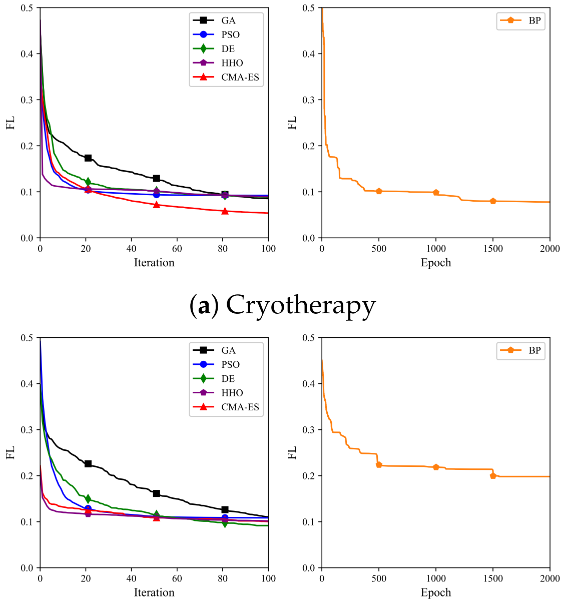
In this study, we attempted to improve the performance of the dendritic neuron model using the CMA-ES. Based on the above tuning results, the performance of the CMA-ES, differential evolution (DE), particle swarm optimization (PSO), genetic algorithm (GA), Harris hawks optimization (HHO) [49], and backpropagation (BP) in training the DNM were compared. To be as fair as possible in the comparisons, the number of iterations of these heuristic algorithms was set to 100 and the epoch of BP was set to 2000. With this configuration, the training time required for the BP algorithm was longer. The convergence curves of the optimization algorithms generated using the MSE function and the FL function are shown in Figures 5 and 6, respectively. The convergence values of each algorithm with the different loss functions are shown in Table 6. The classification accuracies of these optimization algorithms are compared in Table 7.



(**a**) Cryotherapy

(**b**) Immunotherapy

Figure 5. Convergence curves using the MSE for the two datasets.



(**a**) Cryotherapy

(**b**) Immunotherapy

Figure 6. Convergence curves using the FL for the two datasets.

Table 6. Comparison of the convergence results of the six optimization algorithms.

Dataset	Algorithm	Loss Function	
		MSE	**FL**
Cryotherapy	CMA-ES	$3.57 \times 10^{-2} \pm 5.32 \times 10^{-3}$	$5.37 \times 10^{-2} \pm 8.41 \times 10^{-3}$
	DE	$5.35 \times 10^{-2} \pm 6.02 \times 10^{-3}$	$8.86 \times 10^{-2} \pm 6.82 \times 10^{-3}$
	PSO	$4.76 \times 10^{-2} \pm 1.48 \times 10^{-2}$	$9.21 \times 10^{-2} \pm 3.26 \times 10^{-2}$
	GA	$7.59 \times 10^{-2} \pm 1.27 \times 10^{-2}$	$8.54 \times 10^{-2} \pm 3.28 \times 10^{-2}$
	HHO	$7.24 \times 10^{-2} \pm 2.69 \times 10^{-2}$	$8.90 \times 10^{-2} \pm 1.74 \times 10^{-2}$
	BP	$5.55 \times 10^{-2} \pm 2.17 \times 10^{-2}$	$9.33 \times 10^{-2} \pm 4.61 \times 10^{-2}$
Immunotherapy	CMA-ES	$4.69 \times 10^{-2} \pm 8.68 \times 10^{-3}$	$1.00 \times 10^{-1} \pm 2.08 \times 10^{-2}$
	DE	$4.25 \times 10^{-2} \pm 3.76 \times 10^{-3}$	$9.15 \times 10^{-2} \pm 1.25 \times 10^{-2}$
	PSO	$3.87 \times 10^{-2} \pm 1.45 \times 10^{-2}$	$1.08 \times 10^{-1} \pm 2.51 \times 10^{-2}$
	GA	$4.69 \times 10^{-2} \pm 1.05 \times 10^{-2}$	$1.14 \times 10^{-1} \pm 2.62 \times 10^{-2}$
	HHO	$4.24 \times 10^{-2} \pm 1.62 \times 10^{-2}$	$1.01 \times 10^{-1} \pm 2.10 \times 10^{-2}$
	BP	$6.63 \times 10^{-2} \pm 1.83 \times 10^{-2}$	$2.04 \times 10^{-1} \pm 7.70 \times 10^{-2}$

Table 7. Comparison of the classification accuracies of the six optimization algorithms.

Dataset	Loss Function	CMA-ES	DE	PSO	GA	HHO	BP
Cryotherapy	MSE	0.8894	0.8667	0.6649	0.8070	0.8238	0.7904
	FL	0.9012	0.8368	0.6894	0.7403	0.8333	0.8388
Immunotherapy	MSE	0.8404	0.8350	0.7403	0.8157	0.8023	0.8047
	FL	0.8654	0.8526	0.7263	0.7561	0.8357	0.8166

Based on the convergence curves shown in Figures 5a and 6a, the cryotherapy dataset exhibited the fastest convergence rate in terms of both the MSE function and the FL function when using the CMA-ES. For the immunotherapy dataset, as illustrated in Figure 5b, PSO exhibited the fastest convergence rate among all the algorithms. Additionally, Figure 6b demonstrates that the CMA-ES was nearly as fast as the fastest converging DE on the FL function. The convergence results for each algorithm are summarized in Table 6. The CMA-ES achieved the minimum value for the MSE and FL functions on the cryotherapy dataset. For the immunotherapy dataset, PSO reached the minimum convergence value of the MSE function, whereas the CMA-ES almost reached the minimum convergence value of the FL function. These findings indicate that the CMA-ES exhibited powerful optimization performance in training the DNM.

According to the comparison of the classification accuracies of these training algorithms shown in Table 7, the CMA-ES was the best-performing algorithm among the six algorithms because it achieved the highest classification accuracy. In addition, incorporating the FL further improved the prediction accuracy of the CMA-ES, indicating that incorporating the FL as the loss function was necessary.

4.4. Comparison with Classic Machine Learning Models

To further verify the effectiveness of the proposed models, the DNM-FL and DNM-MSE were compared with six popular machine learning models, including multilayer perceptron (MLP), Bayesian classifier (Bayes), support vector machine (SVM), Ada boosting (Ada), K-nearest neighbor (KNN), and decision tree (DT). The hyperparameters of these machine learning models were set as described in Table 8. Six metrics (accuracy, sensitivity, specificity, precision, F1 score value, and AUC value) were used to evaluate each model. The above 6 metrics were averaged from 30 independent experiments. The p-value corresponds to the Wilcoxon signed rank test, which was used to determine the significant differences in accuracy between the DNM-FL and other machine learning models. The confidence level was set at 0.05. Table 9 shows the performance of the eight machine learning models on the two datasets.

Table 8. The hyperparameter settings of the six machine learning models.

Classifier	Parameter	Setting
MLP	Number of layers Number of neurons in hidden layer	3 100
Bayes	Assumption of distribution	Gaussian
SVM	Kernel Penalty parameter	RBF 0.5
Ada	Number of estimators	100
KNN	Number of neighbors	5
DT	Maximum depth	8

Table 9. Comparison of the results of the machine learning models.

Dataset	Classifier	Accuracy	*p*-Value	Sensitivity	Specificity	Precision	F1	AUC
Cryotherapy	MLP	0.8071	0.000015	0.8572	0.8693	0.7830	0.8087	0.8084
	Bayes	0.7023	0.000002	0.7756	0.7633	0.7166	0.7288	0.7943
	SVM	0.8738	0.133519	0.8156	0.8443	0.9429	0.8683	0.9338
	Ada	0.8857	0.503208	0.8863	0.8961	0.9064	0.8914	0.9609
	KNN	0.8119	0.000312	0.7946	0.8002	0.8663	0.8180	0.9004
	DT	0.8464	0.001133	0.8865	0.8867	0.8385	0.8555	0.8452
	DNM-MSE	0.8894	0.411553	0.8651	0.8850	0.9212	0.8900	0.9450
	DNM-FL	0.9012	-	0.8964	0.8919	0.9302	0.9068	0.9630
Immunotherapy	MLP	0.7880	0.000043	0.9930	0.8172	0.7926	0.8797	0.6729
	Bayes	0.7845	0.000012	1.0000	0.8258	0.7845	0.8779	0.5294
	SVM	0.8107	0.002203	0.9845	0.8561	0.8202	0.8925	0.6922
	Ada	0.7928	0.000294	0.8938	0.8461	0.8552	0.8697	0.7759
	KNN	0.7857	0.000026	0.9324	0.8226	0.8269	0.8722	0.7004
	DT	0.8011	0.001825	0.8770	0.8441	0.8725	0.8709	0.7163
	DNM-MSE	0.8404	0.046729	0.9221	0.8437	0.8874	0.9019	0.7770
	DNM-FL	0.8654	-	0.9265	0.8755	0.9101	0.9165	0.7965

Based on the results of the comparison experiments presented in Table 9, it is evident that the DNM-FL achieved superior results on the cryotherapy dataset, with values of 0.9012, 0.8964, 0.8919, 0.9302, 0.9068, and 0.9630 for accuracy, sensitivity, specificity, precision, F1 score, and AUC value, respectively. Ada boosting exhibited the highest specificity of 0.8961 among all the classifiers. Furthermore, the DNM-MSE demonstrated higher accuracy than the other non-DNM classifiers, indicating that the performance of the DNM optimized using the CMA-ES was significantly improved. The DNM-FL outperformed all the compared classifiers in all metrics, except for specificity. As the cryotherapy dataset's samples were relatively balanced, there was no significant difference between the performance of the DNM-FL and DNM-MSE.

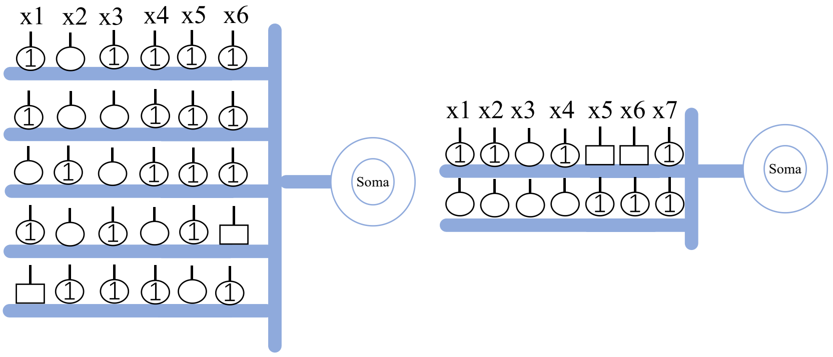
In contrast, on the immunotherapy dataset, which is a typical imbalanced sample dataset, the DNM-FL outperformed the DNM-MSE, achieving values of 0.8654, 0.9265, 0.8755, 0.9101, 0.9165, and 0.7965 for accuracy, sensitivity, specificity, precision, F1 score, and AUC value, respectively. In comparison, the DNM-MSE achieved values of 0.8404, 0.9221, 0.8437, 0.8874, 0.9019, and 0.7770 for the same six metrics. Both the DNM-MSE and DNM-FL outperformed the other six comparison classifiers in all metrics, except for sensitivity. The DNM-FL performed the best among all the comparison classifiers. Importantly, the improvement in specificity reflects the improvement in correctly predicting negative samples. The FL function effectively addressed the issue of the DNM's inability to correctly identify negative samples when the number of negative samples was small.

4.5. Discussion of the Dendritic Pruning Mechanism

The proposed dendritic pruning mechanism was utilized on the trained DNM in this experiment. The classification accuracies of the original DNM and the pruned DNM for 30 independent experiments are compared in Table 10. The typical structures of the pruned DNM for the two datasets are plotted in Figure 7. According to the above-mentioned hyperparameter settings, for the cryotherapy and immunotherapy datasets, the DNM had 12 and 13 dendritic branches before pruning, respectively. For the cryotherapy dataset, the number of dendritic branches of the DNM was reduced from 12 to 5. Similarly, for the immunotherapy dataset, the number of dendritic branches of the DNM was reduced from 13 to 2. The dendritic pruning mechanism is shown to simplify the structures of the DNM, making them more concise. In Table 10, we can see that the accuracy loss of the pruned DNM was less than 0.01, indicating that the proposed dendritic pruning mechanism is effective. The simplified model has a simpler structure and fewer operations.

Table 10. Comparison of the classification accuracies of the original DNM and the pruned DNM.

Dataset	Loss Function	Original DNM	Pruned DNM
Cryotherapy	MSE	0.8894 ± 0.04	0.8841 ± 0.03
	FL	0.9012 ± 0.05	0.8932 ± 0.04
Immunotherapy	MSE	0.8404 ± 0.05	0.8366 ± 0.06
	FL	0.8654 ± 0.03	0.8611 ± 0.04



(a) Cryotherapy (b) Immunotherapy

Figure 7. The typical structures of the pruned DNM for the two datasets.

5. Conclusions

To help in the selection of appropriate treatment methods for patients and improve the accuracy of wart-treatment efficacy prediction, in this study, we constructed a wart-treatment efficacy prediction method based on an improved DNM. The covariance matrix adaptation evolution strategy was combined with the DNM to improve the performance of the DNM while taking into account the interpretability of the optimization process. Due to the sample imbalance in the original dataset, a focal loss function was introduced to address the problem of bias in the generated model toward the majority of samples. Two common datasets of wart-treatment efficacy, the cryotherapy dataset and the immunotherapy dataset, were employed as the benchmark datasets. The proposed CMA-ES-based dendritic neuron model achieved promising results, with average classification accuracies of 0.9012 and 0.8654 on the two datasets, respectively. The superiority of the proposed method was demonstrated by comparing it with six popular machine learning models. Based on the specific pruning mechanism, the structure of the trained DNM can be greatly simplified. The proposed method can help physicians make decisions and is a promising technique that can be integrated into a clinical decision-support system. This study emphasized the importance of artificial intelligence technology in improving medical treatments.

Nevertheless, this study also has the following limitations. First, more datasets of wart-treatment efficacy can be employed to verify the effectiveness of the proposed method. Second, since we do not provide a software suite to implement the DNM, it is not easy to integrate the proposed method into a clinical decision-support system.

In our future work, more comprehensive patient data will be incorporated into the DNM to enhance its generalization ability. Applying the DNM in computer-aided diagnosis will also be a focus of our future efforts.

Author Contributions: Conceptualization, S.S. and B.Z.; methodology, S.S. and B.Z.; software, B.Z. and Q.X.; validation, X.C., Q.X. and J.Q.; formal analysis, B.Z. and Q.X.; investigation, S.S. and J.Q.; resources, S.S. and J.Q.; data curation, S.S. and B.Z.; writing—original draft preparation, S.S. and B.Z.; writing—review and editing, X.C. and J.Q.; visualization, B.Z.; supervision, S.S. and X.C.; project administration, S.S. and J.Q.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 62203069) and the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20220619).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ahmad, T.; Zhang, D.; Huang, C.; Zhang, H.; Dai, N.; Song, Y.; Chen, H. Artificial intelligence in sustainable energy industry: Status Quo, challenges and opportunities. *J. Clean. Prod.* **2021**, *289*, 125834. [CrossRef]
2. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine learning in agriculture: A review. *Sensors* **2018**, *18*, 2674. [CrossRef]
3. Song, S.; Ji, J.; Chen, X.; Gao, S.; Tang, Z.; Todo, Y. Adoption of an improved PSO to explore a compound multi-objective energy function in protein structure prediction. *Appl. Soft Comput.* **2018**, *72*, 539–551. [CrossRef]
4. Chen, X.; Song, S.; Ji, J.; Tang, Z.; Todo, Y. Incorporating a multiobjective knowledge-based energy function into differential evolution for protein structure prediction. *Inf. Sci.* **2020**, *540*, 69–88. [CrossRef]
5. Goecks, J.; Jalili, V.; Heiser, L.M.; Gray, J.W. How machine learning will transform biomedicine. *Cell* **2020**, *181*, 92–101. [CrossRef] [PubMed]
6. Song, S.; Chen, X.; Zhang, Y.; Tang, Z.; Todo, Y. Protein–ligand docking using differential evolution with an adaptive mechanism. *Knowl.-Based Syst.* **2021**, *231*, 107433. [CrossRef]
7. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583–589. [CrossRef]
8. Yu, K.H.; Beam, A.L.; Kohane, I.S. Artificial intelligence in healthcare. *Nat. Biomed. Eng.* **2018**, *2*, 719–731. [CrossRef] [PubMed]
9. Abdar, M.; Książek, W.; Acharya, U.R.; Tan, R.S.; Makarenkov, V.; Pławiak, P. A new machine learning technique for an accurate diagnosis of coronary artery disease. *Comput. Methods Programs Biomed.* **2019**, *179*, 104992. [CrossRef] [PubMed]
10. Yanase, J.; Triantaphyllou, E. A systematic survey of computer-aided diagnosis in medicine: Past and present developments. *Expert Syst. Appl.* **2019**, *138*, 112821. [CrossRef]
11. Jahmunah, V.; Ng, E.; San, T.R.; Acharya, U.R. Automated detection of coronary artery disease, myocardial infarction and congestive heart failure using GaborCNN model with ECG signals. *Comput. Biol. Med.* **2021**, *134*, 104457. [CrossRef]
12. Aamir, M.; Irfan, M.; Ali, T.; Ali, G.; Shaf, A.; Al-Beshri, A.; Alasbali, T.; Mahnashi, M.H. An adoptive threshold-based multi-level deep convolutional neural network for glaucoma eye disease detection and classification. *Diagnostics* **2020**, *10*, 602. [CrossRef] [PubMed]
13. Casal-Guisande, M.; Álvarez Pazó, A.; Cerqueiro-Pequeño, J.; Bouza-Rodríguez, J.B.; Peláez-Lourido, G.; Comesaña-Campos, A. Proposal and Definition of an Intelligent Clinical Decision Support System Applied to the Screening and Early Diagnosis of Breast Cancer. *Cancers* **2023**, *15*, 1711. [CrossRef] [PubMed]
14. Pereira, C.R.; Pereira, D.R.; Weber, S.A.; Hook, C.; De Albuquerque, V.H.C.; Papa, J.P. A survey on computer-assisted Parkinson's disease diagnosis. *Artif. Intell. Med.* **2019**, *95*, 48–63. [CrossRef] [PubMed]
15. Brunetti, A.; Carnimeo, L.; Trotta, G.F.; Bevilacqua, V. Computer-assisted frameworks for classification of liver, breast and blood neoplasias via neural networks: A survey based on medical images. *Neurocomputing* **2019**, *335*, 274–298. [CrossRef]
16. de Souza, R.W.; Silva, D.S.; Passos, L.A.; Roder, M.; Santana, M.C.; Pinheiro, P.R.; de Albuquerque, V.H.C. Computer-assisted Parkinson's disease diagnosis using fuzzy optimum-path forest and Restricted Boltzmann Machines. *Comput. Biol. Med.* **2021**, *131*, 104260. [CrossRef] [PubMed]

17. Aldahan, A.S.; Mlacker, S.; Shah, V.V.; Kamath, P.; Alsaidan, M.; Samarkandy, S.; Nouri, K. Efficacy of intralesional immunotherapy for the treatment of warts: A review of the literature. *Dermatol. Ther.* **2016**, *29*, 197–207. [CrossRef]
18. Salman, S.; Ahmed, M.S.; Ibrahim, A.M.; Mattar, O.M.; El-Shirbiny, H.; Sarsik, S.; Afifi, A.M.; Anis, R.M.; Agha, N.A.Y.; Abushouk, A.I. Intralesional immunotherapy for the treatment of warts: A network meta-analysis. *J. Am. Acad. Dermatol.* **2019**, *80*, 922–930. [CrossRef]
19. Shen, S.; Feng, J.; Song, X.; Xiang, W. Efficacy of photodynamic therapy for warts induced by human papilloma virus infection: A systematic review and meta-analysis. *Photodiagnosis Photodyn. Ther.* **2022**, 102913. [CrossRef]
20. Lechner, M.; Liu, J.; Masterson, L.; Fenton, T.R. HPV-associated oropharyngeal cancer: Epidemiology, molecular biology and clinical management. *Nat. Rev. Clin. Oncol.* **2022**, *19*, 306–327. [CrossRef]
21. Mohammed, G.F.; Al-Dhubaibi, M.S.; Bahaj, S.S.; Elneam, A.I.A. Systemic immunotherapy for the treatment of warts: A literature review. *J. Cosmet. Dermatol.* **2022**, *21*, 5532–5536. [CrossRef] [PubMed]
22. Mulhem, E.; Pinelis, S. Treatment of nongenital cutaneous warts. *Am. Fam. Physician* **2011**, *84*, 288–293. [PubMed]
23. Khozeimeh, F.; Alizadehsani, R.; Roshanzamir, M.; Khosravi, A.; Layegh, P.; Nahavandi, S. An expert system for selecting wart treatment method. *Comput. Biol. Med.* **2017**, *81*, 167–175. [CrossRef] [PubMed]
24. Akben, S.B. Predicting the success of wart treatment methods using decision tree based fuzzy informative images. *Biocybern. Biomed. Eng.* **2018**, *38*, 819–827. [CrossRef]
25. Abdar, M.; Wijayaningrum, V.N.; Hussain, S.; Alizadehsani, R.; Plawiak, P.; Acharya, U.R.; Makarenkov, V. IAPSO-AIRS: A novel improved machine learning-based system for wart disease treatment. *J. Med. Syst.* **2019**, *43*, 220. [CrossRef]
26. Jha, S.K.; Marina, N.; Wang, J.; Ahmad, Z. A hybrid machine learning approach of fuzzy-rough-k-nearest neighbor, latent semantic analysis, and ranker search for efficient disease diagnosis. *J. Intell. Fuzzy Syst.* **2022**, *42*, 2549–2563. [CrossRef]
27. Hu, J.; Ou, X.; Liang, P.; Li, B. Applying particle swarm optimization-based decision tree classifier for wart treatment selection. *Complex Intell. Syst.* **2022**, *8*, 163–177. [CrossRef]
28. Ji, J.; Gao, S.; Cheng, J.; Tang, Z.; Todo, Y. An approximate logic neuron model with a dendritic structure. *Neurocomputing* **2016**, *173*, 1775–1783. [CrossRef]
29. Ji, J.; Song, S.; Tang, Y.; Gao, S.; Tang, Z.; Todo, Y. Approximate logic neuron model trained by states of matter search algorithm. *Knowl.-Based Syst.* **2019**, *163*, 120–130. [CrossRef]
30. Gao, S.; Zhou, M.; Wang, Y.; Cheng, J.; Yachi, H.; Wang, J. Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 601–614. [CrossRef]
31. Luo, X.; Wen, X.; Zhou, M.; Abusorrah, A.; Huang, L. Decision-tree-initialized dendritic neuron model for fast and accurate data classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 4173–7183. [CrossRef] [PubMed]
32. Song, Z.; Tang, Y.; Ji, J.; Todo, Y. Evaluating a dendritic neuron model for wind speed forecasting. *Knowl.-Based Syst.* **2020**, *201*, 106052. [CrossRef]
33. He, H.; Gao, S.; Jin, T.; Sato, S.; Zhang, X. A seasonal-trend decomposition-based dendritic neuron model for financial time series prediction. *Appl. Soft Comput.* **2021**, *108*, 107488. [CrossRef]
34. Tang, C.; Ji, J.; Tang, Y.; Gao, S.; Tang, Z.; Todo, Y. A novel machine learning technique for computer-aided diagnosis. *Eng. Appl. Artif. Intell.* **2020**, *92*, 103627. [CrossRef]
35. Song, S.; Chen, X.; Song, S.; Todo, Y. A neuron model with dendrite morphology for classification. *Electronics* **2021**, *10*, 1062. [CrossRef]
36. Shir, O.M.; Roslund, J.; Whitley, D.; Rabitz, H. Efficient retrieval of landscape Hessian: Forced optimal covariance adaptive learning. *Phys. Rev. E* **2014**, *89*, 063306. [CrossRef] [PubMed]
37. Shir, O.M.; Yehudayoff, A. On the covariance-hessian relation in evolution strategies. *Theor. Comput. Sci.* **2020**, *801*, 157–174. [CrossRef]
38. Hansen, N.; Ostermeier, A. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 20–22 May 1996; IEEE: Piscataway, NJ, USA, 1996; pp. 312–317.
39. Hansen, N.; Müller, S.D.; Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol. Comput.* **2003**, *11*, 1–18. [CrossRef]
40. Hansen, N. The CMA evolution strategy: A tutorial. *arXiv* **2016**, arXiv:1604.00772.
41. Haixiang, G.; Yijing, L.; Shang, J.; Mingyun, G.; Yuanyue, H.; Bing, G. Learning from class-imbalanced data: Review of methods and applications. *Expert Syst. Appl.* **2017**, *73*, 220–239. [CrossRef]
42. Wang, L.; Han, M.; Li, X.; Zhang, N.; Cheng, H. Review of classification methods on unbalanced data sets. *IEEE Access* **2021**, *9*, 64606–64628. [CrossRef]
43. Prusa, J.; Khoshgoftaar, T.M.; Dittman, D.J.; Napolitano, A. Using random undersampling to alleviate class imbalance on tweet sentiment data. In Proceedings of the 2015 IEEE International Conference on Information Reuse and Integration, Washington, DC, USA, 13–15 August 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 197–202.
44. Luo, Z.; Parvïn, H.; Garg, H.; Qasem, S.N.; Pho, K.; Mansor, Z. Dealing with imbalanced dataset leveraging boundary samples discovered by support vector data description. *Comput. Mater. Contin.* **2021**, *66*, 2691–2708. [CrossRef]
45. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

46. Khozeimeh, F.; Jabbari Azad, F.; Mahboubi Oskouei, Y.; Jafari, M.; Tehranian, S.; Alizadehsani, R.; Layegh, P. Intralesional immunotherapy compared to cryotherapy in the treatment of warts. *Int. J. Dermatol.* **2017**, *56*, 474–478. [CrossRef] [PubMed]
47. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
48. Karna, S.K.; Sahai, R. An overview on Taguchi method. *Int. J. Eng. Math. Sci.* **2012**, *1*, 1–7.
49. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.