

Article

Introducing an Evolutionary Algorithm for the Optimal Training of RBF Networks

Ioannis G. Tsoulos ^{1,*}, Vasileios Charilogis ¹ and Dimitrios Tsalikakis ²¹ Department of Informatics and Telecommunications, University of Ioannina, 45110 Ioannina, Greece; v.charilog@uoi.gr² Department of Engineering Informatics and Telecommunications, University of Western Macedonia, 50100 Kozani, Greece; dtsalikakis@uowm.gr

* Correspondence: itsoulos@uoi.gr

Abstract

A large collection of real-world classification and regression problems can be addressed using machine learning tools such as, for example, radial basis function networks (RBF networks). However, the techniques used for training RBF networks often exhibit various problems, such as getting trapped in the local minima of the error function, or even encountering numerical issues when solving systems of linear equations in order to estimate the parameters of the RBF network. This paper presents a multi-stage evolutionary technique based on genetic algorithms for the effective training of RBF networks. In the first stage, the value ranges of the RBF network parameters are estimated using the K-Means algorithm. In the second stage, the chromosomes of the genetic algorithm are initialized within the parameter ranges determined in the first stage, followed by the execution of the genetic algorithm. Each chromosome of the genetic algorithm is considered a candidate parameter vector for the machine learning model. The centers and variances of the RBF network are estimated by the genetic algorithm, while the network weights are determined by solving a system of linear equations. This method was applied to a large set of classification and data-fitting problems, yielding excellent results.

Keywords: machine learning; neural networks; genetic algorithms**MSC:** 65K05; 65K10; 68W01

1. Introduction

The rapid development of machine learning has enabled its adoption in numerous practical applications. Representative examples can be found in scientific and technological fields such as physics [1,2], astronomy [3,4], chemistry [5,6], medicine [7,8], economics [9,10], image processing [11], time-series prediction [12], financial and credit card analysis [13], and mechanical engineering [14]. Such problems are typically tackled through machine learning approaches, with Radial Basis Function (RBF) networks [15] representing one of the most widely adopted and effective model classes. An RBF network is commonly defined as a function:

$$R(\vec{x}) = \sum_{i=1}^k w_i \phi(\|\vec{x} - \vec{c}_i\|) \quad (1)$$

The following notation is used in Equation (1):

1. The vector $\vec{x} = (x_1, x_2, \dots, x_d)$ denotes the input pattern.



Academic Editor: Zhizhong Liu

Received: 27 June 2026

Revised: 31 July 2026

Accepted: 6 August 2026

Published: 7 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)[Attribution \(CC BY\)](#) license.

2. The constant k denotes the number of weights for the RBF network.
3. The vectors $\vec{c}_i$, $i = 1, \dots, k$ stand for the so-called center vectors.
4. The value $R(\vec{x})$ denotes the predicted value of the network for the input pattern $\vec{x}$.
5. The function $\phi(x)$ typically is defined as the following Gaussian function:

$$\phi(x) = \exp\left(-\frac{(x - c)^2}{\sigma^2}\right) \quad (2)$$

6. The vector $\vec{w}$ represents the weight vector of the RBF network. The vector $\vec{w}$ contains the coefficients used to weight the output of each basis function $\phi(x)$.

Moreover, the training error associated with any RBF network $R(x)$ is defined as:

$$E(R(x)) = \sum_{i=1}^M (R(\vec{x}_i) - y_i)^2 \quad (3)$$

In this equation, the set $(\vec{x}_i, y_i)$, $i = 1, \dots, M$ represents the training set of the objective problem and the values y_i are the actual outputs for any given pattern $\vec{x}_i$. The effectiveness of RBF networks has been demonstrated across numerous application domains. Representative examples include physics-related problems [16–19], differential equation solving [20–22], robotic systems [23,24], facial recognition [25], digital communication systems [26,27], chemical applications [28,29], economic modeling [30–32], cybersecurity and network security tasks [33], time-series forecasting [34], wind energy production prediction [35], constrained optimization [36,37], etc. A schematic description of the RBF network is outlined in Figure 1.

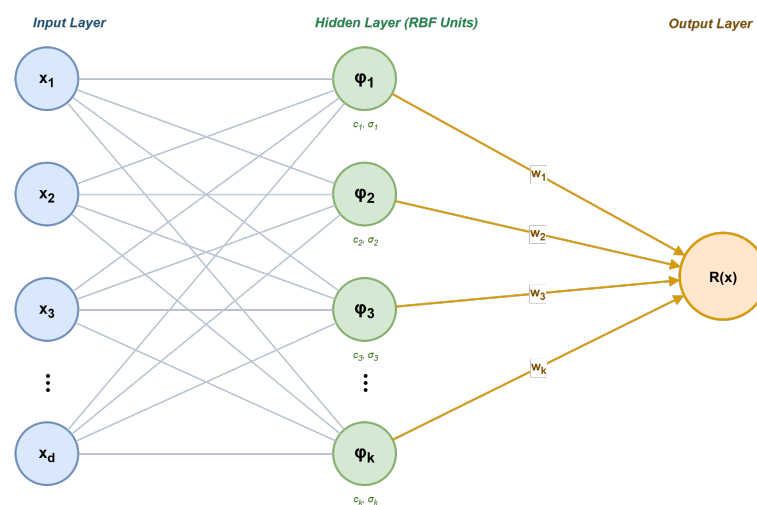


Figure 1. General structure of an RBF network. The input vector $\vec{x}$ is fed to k hidden units, each characterized by a center $\vec{c}_i$ (a reference point in the input space) and a variance σ_i (the spread of the basis function). The output $R(x)$ is a weighted sum of the hidden unit responses, with weights w_i .

In recent years, numerous studies have introduced novel approaches for initializing the parameters of RBF networks [38–40]. In addition, the influence of kernel widths on the performance of RBF networks has been extensively investigated by Benoudjit et al. [41], while Neruda et al. [42] provided a comparative evaluation of several learning algorithms designed for RBF training. Furthermore, a variety of pruning strategies have been proposed [43–45] with the objective of reducing the number of network parameters and, consequently, the overall model complexity. Given the widespread adoption of RBF networks and the substantial computational effort often required for their training, several studies have recently focused on exploiting parallel computing architectures to accelerate

neural network parameter optimization [46,47]. Finally, Park et al. [48] demonstrated that an RBF network consisting of a single hidden processing layer possesses the universal approximation property, being capable of approximating any continuous function with arbitrary accuracy.

Moreover, Mohammadi et al. proposed [49] a novel hybrid evolutionary-based RBF network method for forecasting time series that can estimate both the network architecture as well as the values for the parameters of the network. Likewise, Wu et al introduced [50] a hybrid method that combines the Particle Swarm Optimization (PSO) [51] method as well as genetic algorithms to determine the parameters of RBF networks. Recently, Elansari proposed a new method [52] that utilizes genetic algorithms for the optimal training of RBF networks. Likewise, Mukherjee et al. developed hybrid Gaussian Radial Basis Neural Networks (GRAB-NNs) for data-driven models in a recent study [53]. Beyond genetic algorithms, other population-based metaheuristics have also been applied to RBF training, including differential evolution [54] and particle swarm optimization [55,56].

A third research direction recognizes that purely global metaheuristics, while effective at exploration, converge slowly in the vicinity of the optimum, motivating hybrid or memetic algorithms that combine global search with local refinement [57,58]. Recent RBF-specific examples include hybrid schemes combining particle swarm optimization, cuckoo search, and biogeography-based optimization, refined by a local BFGS step [59].

In conventional RBF network training, the centers of the basis functions $\phi(x)$ are initially determined, and the corresponding weight vector $\vec{w}$ is subsequently obtained by solving a linear system of equations. The k-means clustering algorithm [60] is among the most frequently employed techniques for center selection. Despite its simplicity and computational efficiency, this approach often suffers from overfitting, reducing the predictive performance of the model on unseen samples. Moreover, because the optimization process does not impose explicit constraints on the parameter values, the resulting estimates may become excessively large or small, adversely affecting both model stability and generalization capability.

The present work introduces a novel training methodology for RBF networks consisting of two distinct phases. During the first phase, a promising range of values for the network centers and variances is estimated by exploiting the k-means clustering technique. In the second phase, a modified genetic algorithm [61,62] is employed to determine the optimal values of the centers and variances within the search intervals identified during the first phase. The weights of the RBF network are computed through the solution of an appropriate linear system of equations.

The present work is positioned at the intersection of all three research directions described above: the first phase draws on the initialization literature by using K-means to establish informed, data-driven search bounds rather than fixed or arbitrary ranges; the second phase contributes a genetic algorithm operating within these bounds, following the evolutionary training paradigm; and the local optimization step applied at the end of the second phase follows the hybrid/memetic paradigm. Unlike prior evolutionary RBF methods that treat the search space as unconstrained, and unlike prior hybrid methods that do not exploit data-driven initialization to constrain the search space before the global search begins, the core contribution of the proposed method lies in this three-stage integration.

The main contributions of the proposed method are summarized as follows:

1. The k-means clustering algorithm is employed to effectively identify suitable value ranges for the parameters of the RBF network.
2. A well-established global optimization technique, namely genetic algorithms (GAs), is introduced to efficiently determine the optimal parameters of an RBF network, regardless of the dataset to which it is applied.

3. A numerical method is proposed for estimating the output weights of the RBF network.
4. A local optimization procedure is incorporated to further reduce the training error of the RBF network.

The remaining of this article is organized as follows: in Section 2 the proposed method is fully described, in Section 3 the experimental results are outlined and finally in Section 4 a discussion is provided on the experimental results.

2. Materials and Methods

2.1. The First Phase of the Proposed Method

In the first phase, the k-means clustering algorithm is employed to estimate promising search intervals for the centers and variances of the RBF network. Specifically, for each network parameter, the corresponding search interval is defined as a multiple of the value produced by the k-means algorithm, thereby providing a bounded region for the subsequent optimization process. The k-means algorithm is presented in Algorithm 1. This algorithm is an unsupervised clustering technique that splits a dataset into a number of predefined clusters by minimizing the within-cluster sum of squared distances. The algorithm iteratively assigns each data point to its nearest cluster centroid and at every generation updates the centroid as the mean of the assigned points. This process is repeated until convergence.

The set of parameters comprising the centers and variances of an RBF network with k hidden units is represented by a vector of dimension

$$n = k \times d + k \quad (4)$$

where d denotes the dimensionality of the input patterns. This vector has the following structure:

$$\vec{T} = \{c_{11}, c_{12}, c_{1d}, \sigma_1, c_{21}, c_{22}, \dots, c_{2d}, \sigma_2, \dots, c_{k1}, c_{k2}, \dots, c_{kd}, \sigma_k\} \quad (5)$$

For this vector of parameters, the bound vectors $\vec{L}$, $\vec{R}$ are constructed using the procedure depicted in Algorithm 2.

2.2. The Description of the Second Phase

In the second phase of the proposed algorithm, a genetic algorithm is employed to optimize the centers and variances of the RBF network within the search intervals defined by the vectors $\vec{L}$, $\vec{R}$ obtained during the first phase. The output weight vector $\vec{w}$ of the RBF network is subsequently computed by solving the corresponding linear system of equations. Within the proposed genetic algorithm, each chromosome follows the structure of the parameter vector $\vec{T}$ introduced in Equation (5), where all decision variables corresponding to the RBF centers and variances are encoded.

A suitably modified genetic algorithm was adopted in this stage because genetic algorithms constitute robust evolutionary optimization techniques that have demonstrated their effectiveness across a broad spectrum of practical applications. Genetic algorithms (GAs) are evolutionary optimization techniques designed to search for the global minimum of a multidimensional objective function. To achieve this, a GA maintains a population of candidate solutions, referred to as chromosomes. These chromosomes evolve continuously over successive iterations of the algorithm, known as generations, through the application of genetic operators that mimic natural evolutionary processes, including selection, crossover, and mutation. In addition, they are particularly well-suited for integration with machine learning models, as they generally require only the formulation of a suitable fitness function while treating the underlying model as a black-box optimization problem. For example,

they have been applied to energy problems [63], water distribution problems [64], economic problems [65], training of neural networks [66], statistics [67], RBF interpolation [68], etc. The steps of the used genetic algorithm have as follows:

Algorithm 1 The k-means algorithm used to estimate centers and variances.

1. **Initialization**

- (a) **Input:** the training set $(\vec{x}_i, y_i)$, $i = 1, \dots, M$ of the objective problem.
- (b) **Set** as k the number of desired centers.
- (c) **Assign** randomly every pattern x_i , $i = 1, \dots, M$ in a cluster S_j , $j = 1, \dots, k$.
- (d) **For** each cluster $j = 1, \dots, k$ **do**
 - i. **Denote** with M_j the number of points in S_j
 - ii. **Estimate** every center c_j as

$$\vec{c}_i = \frac{1}{M_j} \sum_{\vec{x}_i \in S_j} \vec{x}_i$$

- (e) **End For**

2. **Repeat**

- (a) Set $S_j = \{\}$, $j = 1, \dots, k$
- (b) **For** $i = 1, \dots, M$ **do**
 - i. **Set** $j^* = \operatorname{argmin}_{m=1}^k \{D(x_i, c_m)\}$. In this formulation, j^* denotes the index of the closest center to the input sample $\vec{x}_i$, whereas $D(x, y)$ corresponds to the Euclidean distance function.
 - ii. **Set** $S_{j^*} = S_{j^*} \cup \{x_i\}$.
- (c) **End For**
- (d) **For** $j = 1, \dots, k$ **do**
 - i. **Calculate** the number of points in each cluster S_j and denote them as M_j .
 - ii. **Calculate** every center $\vec{c}_i$ using the equation:

$$\vec{c}_i = \frac{1}{M_j} \sum_{\vec{x}_i \in S_j} \vec{x}_i$$

- iii. **Calculate** the quantities σ_j as

$$\sigma_j^2 = \frac{\sum_{i=1}^{M_j} (x_i - c_j)^2}{M_j}$$

- (e) **End For**

- 3. **If** the centers $\vec{c}_i$, $j = 1, \dots, k$ did not change then the loop is terminated.
 - 4. **End Repeat**
 - 5. The algorithm **returns** as output the estimated centers $\vec{c}_i$, $i = 1, \dots, k$ and the associated variances σ_i , $i = 1, \dots, k$
-

1. **Initialization step.**

- (a) The algorithm has the following set of parameters:
 - i. N_c , the desired number of chromosomes.
 - ii. N_g , the maximum number of allowed generations.
 - iii. p_s , the used selection rate with $p_s \leq 1$.
 - iv. p_m , the used mutation rate with $p_m \leq 1$.

- (b) **Construct** a set of chromosomes with dimension n as defined in Equation (4). Each chromosome represents the centers and variances of any given RBF network and the value of each chromosome is initialized inside the bound vectors $\vec{L}$, $\vec{R}$.
 - (c) **Set** it = 0, which represents the current number of generations.
2. **Fitness calculation step.**

- (a) **For** $i = 1, \dots, N_c$ **do**
 - i. **Produce** an RBF network $R_i = R(\vec{x})$. The centers and variances of R_i are derived from chromosome $\vec{g}_i$. The weight vector $\vec{w} = (w_1, w_2, \dots, w_k)$ is calculated as follows:
 - A. **Set** $W = [w_1, w_2, \dots, w_k]$
 - B. **Set** $\Phi = \phi_j(x_i)$, that denotes the outputs of the computation units.
 - C. **Set** $T = \{t_j = y_j, j = 1, \dots, M\}$, that denotes the real outputs for the training set.
 - D. **Solve** the following system of equations:

$$\Phi^T (T - \Phi W^T) = 0 \quad (6)$$

The solution is

$$W^T = (\Phi^T \Phi)^{-1} \Phi^T T = \Phi^\dagger T \quad (7)$$

- ii. **Calculate** the corresponding fitness value f_i as

$$f_i = \sum_{j=1}^M (R_i(\vec{x}_j) - y_j)^2 \quad (8)$$

- (b) **End For**
3. **Genetic operations step.**
- (a) **Selection** procedure: The population is first ordered based on fitness. A fraction p_s of the population, corresponding to the top $p_s \times N_c$ chromosomes, is retained intact for the next generation. The rest of the population is regenerated by applying crossover and mutation operations to selected parent chromosomes.
 - (b) **Crossover** procedure: In this step $(1 - p_s) \times N_c$ new offsprings will be produced. For each pair $(\tilde{z}, \tilde{w})$ of these new chromosomes, two chromosomes (z, w) are chosen from the current population with the assistance of tournament selection. The new chromosomes will be created using the following equations:

$$\begin{aligned} \tilde{z}_i &= a_i z_i + (1 - a_i) w_i \\ \tilde{w}_i &= a_i w_i + (1 - a_i) z_i \end{aligned} \quad (9)$$

In this equation, a_i denotes random numbers in the range $[-0.5, 1.5]$ [69].

- (c) **Mutation** procedure: Select a random number $r \in [0, 1]$ for each element $\vec{g}_{i,j}, j = 1, \dots, n$ for the corresponding chromosome $\vec{g}_i$. If $r \leq p_m$, then this element is changed using the following scheme:

$$\vec{g}_{i,j} = \vec{g}_{i,j} + t \times p_c \times \vec{g}_{i,j} \quad (10)$$

where p_c is a number in range $[0, 1]$ that denotes the magnitude of changes applied to the current element and t is randomly selected integer that could be -1 or 1 .

4. **Termination check step.**

- (a) **Set** $it = it + 1$
- (b) **If** $it < N_g$ then **goto** Fitness Calculation step.
- (c) **Otherwise**, obtain the best chromosome $\vec{g}^*$ and create the associated RBF network $R^* = R(\vec{x})$. The weight vector for this RBF network is also obtained by solving a similar system of equations as in Fitness Calculation Step.
- (d) **Apply** a local search procedure to minimize the error function:

$$f^* = \sum_{j=1}^M (R^*(\vec{x}_j) - y_j)^2 \quad (11)$$

A modified BFGS algorithm based on the method of Powell [70] was adopted as the local optimization procedure. This variant efficiently enforces the parameter bounds established during the preceding stage of the proposed methodology.

Algorithm 2 The algorithm that is used to construct the bound vectors $\vec{L}$, $\vec{R}$.

1. **Input:** the vector $\vec{I}$ of Equation (5).
 2. **Input:** the scale factor used for the bound vectors denoted as F with $F \geq 1$.
 3. **Input:** a small positive number v_e used as the lower limit of the variables representing the variances of the RBF network.
 4. **Set** $m = 1$
 5. **For** $i = 1, \dots, k$ **do**
 - (a) **For** $j = 1, \dots, d$ **do** //this loop is for the variables of each center
 - i. **Set** $L_m = -F \times |I_m|$ //this denotes the m element of vector $\vec{L}$.
The value I_m represents the m element of vector $\vec{I}$.
 - ii. **Set** $R_m = F \times |I_m|$ //this denotes the m element of vector $\vec{R}$.
 - iii. **Set** $m = m + 1$
 - (b) **End For**
 - (c) **Set** $L_m = v_e$ //set the bounds for the variables representing the variances.
 - (d) **Set** $R_m = F \times |I_m|$
 - (e) **Set** $m = m + 1$
 6. **End For**
 7. **Output:** The bound vectors $\vec{L}$, $\vec{R}$
-

The overall structure of the second phase is illustrated in Figure 2. The procedure begins with the initialization of a population of chromosomes within the parameter bounds established during the first phase. The centers and variances of each candidate RBF network are extracted directly from the corresponding chromosome, the network weights are subsequently estimated by solving the corresponding linear system of equations, and a fitness value is assigned to each candidate based on the resulting prediction error. Following the evaluation of all candidates, the genetic operations of selection, crossover, and mutation are applied to produce the next generation. The generation counter is then incremented and checked against the maximum allowed number of generations. This cycle repeats until the termination criterion is met. At that point, the best chromosome identified throughout the search is used to reconstruct the final RBF network, whose weights are re-estimated

by solving the linear system once more, after which a local optimization step based on the BFGS method is applied to further refine the solution.

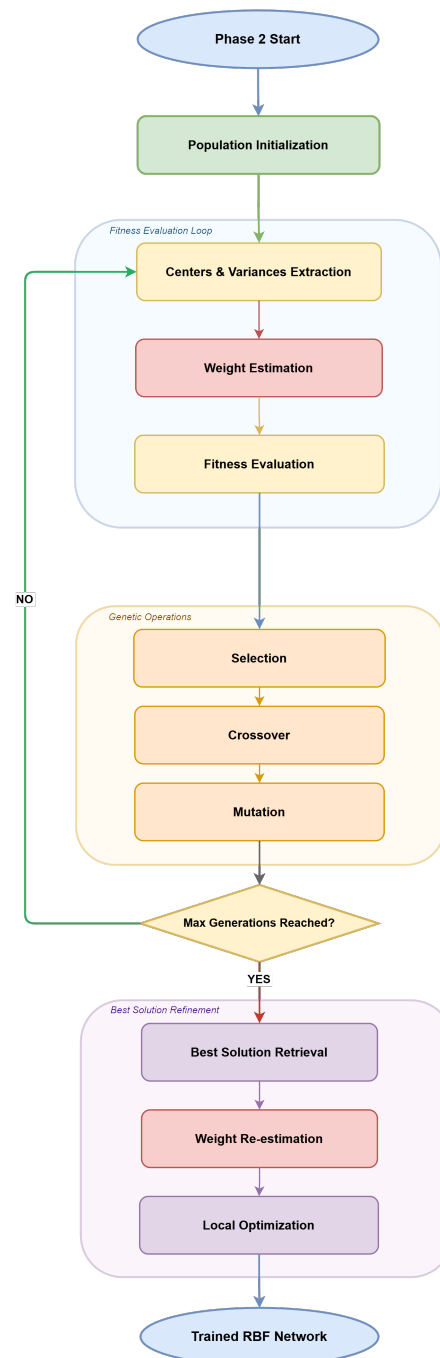


Figure 2. Flowchart of the second phase of the proposed training procedure.

3. Results

3.1. Experimental Datasets

The datasets used in the conducted experiments were obtained from the following online databases:

1. The UCI website, <https://archive.ics.uci.edu/> (accessed on 24 June 2026) [71]
2. The Keel website, <https://sci2s.ugr.es/keel/datasets.php> (accessed on 24 June 2026) [72].
3. The Statlib URL <ftp://lib.stat.cmu.edu/datasets/index.html> (accessed on 24 June 2026).

The list of classification datasets has as follows:

1. **Appendictis** dataset, which is a medical dataset [73].
2. **Alcohol** dataset, related to alcohol consumption and appeared in [74].
3. **Australian** dataset [75] used in economics.
4. **Balance** dataset [76], related to various psychological experiments.
5. **Bands** dataset, used to detect printing problems [77].
6. **Circular** dataset, which is produced artificially and contains two distinct classes.
7. **Cleveland** dataset, related to the presence of heart disease [78,79].
8. **Dermatology** dataset [80], a medical dataset.
9. **Ecoli**, used to detect protein problems [81].
10. The **Fert** dataset, related to sperm concentration problems.
11. **Glass** dataset, related to glass issues.
12. **Haberman**, a medical dataset that contains data used in the prediction of breast cancer.
13. **Hayes roth** dataset [82].
14. **Heart** dataset [83], which was used to detect heart diseases.
15. **HeartAttack**, a medical dataset related to heart diseases.
16. **HouseVotes** dataset [84].
17. **Ionosphere** dataset, that was used to classify radar data from the ionosphere [85,86].
18. **Liverdisorder** dataset [87], which is a medical dataset.
19. **Lymography** dataset [88].
20. **Mammographic** dataset [89], which is a medical dataset related to breast tumors.
21. **Parkinsons** dataset, which is a medical dataset used to detect the presence of the Parkinson's Disease [90].
22. **Page Blocks** dataset, related to the determination of blocks inside various documents.
23. **Phoneme**, a dataset related to sound.
24. **Pima** dataset, which is a medical dataset used widely [91].
25. **Popfailures** dataset [92], used in various experiments regarding climate.
26. **Ring**, a dataset related to some distributions.
27. **Spiral** dataset, which was produced artificially and contains two classes.
28. **Regions2** dataset [93].
29. **Saheart** dataset [94], a medical dataset related to the presence of heart diseases.
30. **Satimage** dataset, related to satellite images.
31. **Segment** dataset [95], which is an image processing dataset.
32. The **Sonar** dataset [96].
33. **Spambase** dataset, regarding the detection of spam emails [97].
34. **Statheart**, which is medical dataset on heart diseases.
35. **Student**, which contains data from various experiments in schools [98].
36. **Transfusion** dataset [99].
37. **Wdbc** dataset [100], which is a medical dataset related the presence of breast tumors.
38. **Wine** dataset, used for the classification of the quality of wines [101,102].
39. **Eeg** dataset, which is an EEG dataset [103]. From this dataset the following cases were used: Z_F_S, ZONF_S and ZO_NF_S.
40. **Zoo** dataset [104], which was used to estimate the class of various animals.

In addition, the list of the used regression datasets has as follows:

1. **Abalone**, used to detect the age of abalones [105].
2. **Airfoil**, a dataset derived from NASA [106].
3. **Auto**, used to estimate the fuel consumption of cars.
4. **BK**, used to estimate the points scored in basketball games.
5. **BL**, that contains measurements from electricity experiments.
6. **Baseball**, used for the estimation of income of baseball players.

7. **Concrete**, a dataset that contains data from civil engineering [107].
8. **DEE**, a dataset related to the prediction of the prices of electricity.
9. The **FA** dataset, which is related to fat measurements.
10. **FY**, that contains measurements related to fruit flies.
11. **HO**, which was downloaded from the STATLIB repository.
12. **Housing**, a dataset that contains data from prices of houses [108].
13. **Laser**, a dataset containing data from various physics experiments.
14. **LW**, a dataset used to detect problems in the weight of babes.
15. **MB** dataset [109].
16. **Mortgage**, a dataset that contains economic data.
17. **NT** dataset [110].
18. **PL** dataset, which was downloaded from the STATLIB repository.
19. **Plastic**, a dataset that contains measurements from plastic experiments.
20. **PY** dataset [111].
21. **Quake**, related to measurements on quakes [112].
22. **SN**, which is a benchmark dataset obtained from the STATLIB repository.
23. **Stock**, related to the prices of stocks.
24. **Treasury**, which is a dataset contains economic data.
25. The **VE** dataset, which was downloaded from the STATLIB repository.

3.2. Experimental Details

The RBF network implemented in this study was developed in ANSI C++ with the support of the open-source Armadillo v15.4 library [113]. The optimization algorithms used were obtained from the OPTIMUS v1.0.0 environment, downloaded from <https://github.com/itsoulos/OPTIMUS/> (accessed on 24 June 2026) [114]. Furthermore, derivative computations required by the optimization procedures were performed using the Adept v2.1.1 automatic differentiation library [115]. All the experiments were conducted on an Linux system with 128 GB of RAM. During the experimental evaluation, the 10-fold cross-validation technique was employed, where each dataset was partitioned into ten folds. In each iteration, 90% of the instances were used as the training set, while the remaining 10% constituted the test set. Both the training procedure and the application of the k-means initialization method were performed exclusively on the training set, with no information from the test set being used during model construction. Each experimental setup was repeated 30 independent times, with a distinct initialization seed assigned to the random number generator in every run. Random number generation was carried out using the drand48() function provided by the C programming language. The average classification error is defined as

$$E_C(N(\vec{x}, \vec{w})) = 100 \times \frac{\sum_{i=1}^N (\text{class}(N(\vec{x}_i, \vec{w})) \neq y_i)}{M} \quad (12)$$

The set T denotes the corresponding test set, with $T = (x_i, y_i)$, $i = 1, \dots, M$. Also, the average regression error is defined as follows:

$$E_R(N(\vec{x}, \vec{w})) = \frac{\sum_{i=1}^N (N(\vec{x}_i, \vec{w}) - y_i)^2}{M} \quad (13)$$

The settings used in the conducted experiments are listed in Table 1.

Table 1. Experimental settings.

Parameter	Meaning	Value
N_c	Number of chromosomes finalstrutarstrutbox	500
N_g	Maximum number of generations finalstrutarstrutbox	500
p_s	Selection rate	0.10
p_m	Mutation rate	0.05
p_c	Mutation magnitude	0.05
F	Scale factor	4.0
v_e	Small value used in variances finalstrutarstrutbox	0.01
k	Number of weights	10

Also, in the tables containing experimental results, the following notation is used:

1. The column DATASET represents the name of the used dataset.
2. The ADAM column stands for the usage of the ADAM optimizer [116,117] to train a neural network with 10 processing nodes.
3. The column BFGS stands for the results obtained when the BFGS optimization procedure [118] is used to train an artificial neural network [119,120] with 10 processing nodes.
4. The column NEAT [121] is used to denote the results from NEAT (NeuroEvolution of Augmenting Topologies).
5. The column RBF represents the results where the traditional training method is used to train an RBF network with 10 processing nodes.
6. The column GRBF is used for the experimental results of the method presented in [122] to train an RBF network with 10 processing nodes.
7. The column RBFCON denotes the incorporation of a recently introduced method [123] that constructs RBF networks with the assistance of Grammatical Evolution [124].
8. The column PROPOSED stands for the results obtained by the incorporation of the proposed method.
9. The row labeled AVERAGE presents the mean classification or regression error obtained across the entire collection of datasets.

3.3. Comparison with Other Machine Learning Methods

The results of the comparative evaluation are presented in Table 2 for the classification problems and in Table 3 for the regression problems, while Figures 3 and 4 display the distribution of errors via box plots and report the outcomes of the overall Friedman test. For classification tasks (Table 2), the mean error rates were 35.18% for ADAM, 35.50% for GRBF, 33.25% for BFGS, 32.16% for NEAT, 28.68% for RBF, 24.96% for RBFCON, and 21.58% for the proposed method. The overall Friedman test (Figure 3) yielded $p = 3.60 \times 10^{-16}$, confirming that the differences among methods are statistically significant at a very high confidence level. The proposed method achieves the lowest mean error and outperforms the second-best method, RBFCON, by approximately 13.5%. At the individual dataset level, the proposed method ranked first in 21 out of 42 datasets (50.0%), with particularly strong performance on RING, SPAMBASE, Z_F_S, and ZONF_S, as detailed in Table 2.

Table 2. Experimental results on the classification datasets, using a variety of machine learning methods.

Dataset	ADAM	BFGS	NEAT	RBF	GRBF	RBFCON	Proposed
APPENDICITIS	16.50%	18.00%	17.20%	12.23%	16.83%	13.60%	17.80%
ALCOHOL	57.78%	41.50%	66.80%	49.38%	52.45%	51.24%	20.43%
AUSTRALIAN	35.65%	38.13%	31.98%	34.89%	41.79%	14.14%	18.90%
BALANCE	12.27%	8.64%	23.14%	33.53%	38.02%	39.94%	17.50%
BANDS	36.92%	36.67%	34.30%	37.17%	43.95%	35.75%	36.28%
CIRCULAR	19.95%	6.08%	35.18%	5.98%	21.43%	18.26%	5.04%
CLEVELAND	67.55%	77.55%	53.44%	67.10%	67.47%	49.14%	50.14%
DERMATOLOGY	26.14%	52.92%	32.43%	62.34%	61.46%	45.20%	39.80%
ECOLI	64.43%	69.52%	43.44%	59.48%	70.58%	54.18%	58.36%
FERT	23.98%	23.20%	15.37%	15.00%	15.00%	15.20%	21.40%
GLASS	61.38%	54.67%	55.71%	50.05%	66.91%	55.38%	52.76%
HABERMAN	29.00%	29.34%	24.04%	25.10%	24.37%	26.27%	27.37%
HAYES ROTH	59.70%	37.33%	50.15%	64.36%	63.46%	34.54%	40.08%
HEART	38.53%	39.44%	39.27%	31.20%	28.44%	17.22%	22.00%
HEARTATTACK	45.55%	46.67%	32.34%	29.00%	40.48%	22.60%	19.13%
HOUSEVOTES	7.48%	7.13%	10.89%	6.13%	11.99%	3.05%	4.61%
IONOSPHERE	16.64%	15.29%	19.67%	16.22%	19.83%	14.32%	12.83%
LIVERDISORDER	41.53%	42.59%	30.67%	30.84%	36.97%	32.21%	30.41%
LYMOGRAPHY	39.79%	35.43%	33.70%	25.50%	29.33%	26.36%	24.57%
MAMMOGRAPHIC	46.25%	17.24%	22.85%	21.38%	30.41%	17.67%	17.72%
PARKINSONS	24.06%	27.58%	18.56%	17.41%	33.81%	12.79%	15.53%
PAGE BLOCKS	34.27%	8.49%	10.22%	10.09%	10.83%	11.09%	9.47%
PHONEME	29.43%	15.58%	22.34%	23.32%	27.70%	24.58%	15.59%
PIMA	34.85%	35.59%	34.51%	25.78%	27.83%	24.13%	23.59%
POPFAILURES	5.18%	5.24%	7.05%	7.04%	7.08%	6.98%	5.61%
REGIONS2	29.85%	36.28%	33.23%	38.29%	39.98%	26.34%	25.71%
RING	28.80%	29.44%	30.85%	21.67%	52.03%	11.08%	5.05%
SAHEART	34.04%	37.48%	34.51%	32.19%	33.90%	29.52%	28.89%
SATIMAGE	73.22%	71.73%	76.19%	40.19%	61.20%	59.16%	32.97%
SEGMENT	49.75%	68.97%	66.72%	59.68%	54.25%	62.36%	32.47%
SONAR	30.33%	25.85%	34.10%	27.90%	37.13%	27.20%	24.00%
SPAMBASE	48.05%	18.16%	35.77%	29.35%	35.36%	16.95%	10.54%
SPIRAL	47.67%	47.99%	48.66%	44.87%	50.02%	42.14%	41.03%
STATHEART	44.04%	39.65%	44.36%	31.36%	42.94%	19.22%	23.89%
STUDENT	5.13%	7.14%	10.20%	5.49%	33.26%	7.63%	12.43%
TRANSFUSION	25.68%	25.84%	24.87%	26.41%	25.67%	25.15%	25.23%
WDBC	35.35%	29.91%	12.88%	7.27%	8.82%	6.77%	4.19%
WINE	29.40%	59.71%	25.43%	31.41%	31.47%	11.00%	10.59%
Z_F_S	47.81%	39.37%	38.41%	13.16%	23.37%	11.13%	4.68%
ZO_NF_S	47.43%	43.04%	43.75%	9.02%	22.18%	11.08%	5.47%
ZONF_S	11.99%	15.62%	5.44%	4.03%	17.41%	4.14%	1.82%
ZOO	14.13%	10.70%	20.27%	21.93%	33.50%	11.60%	10.60%
Average	35.18%	33.25%	32.16%	28.68%	35.50%	24.96%	21.58%

Post hoc pairwise comparisons (Conover test, Holm-corrected) confirm that the proposed method differs significantly ($p_{\text{adj}} < 0.05$) from RBF ($p_{\text{adj}} = 0.0276$), NEAT ($p_{\text{adj}} = 5.39 \times 10^{-6}$), BFGS ($p_{\text{adj}} = 2.31 \times 10^{-6}$), ADAM ($p_{\text{adj}} = 2.06 \times 10^{-8}$), and GRBF ($p_{\text{adj}} = 7.76 \times 10^{-11}$), but does not differ significantly from RBFCON ($p_{\text{adj}} = 0.7358$), indicating statistically comparable classification performance between the two methods.

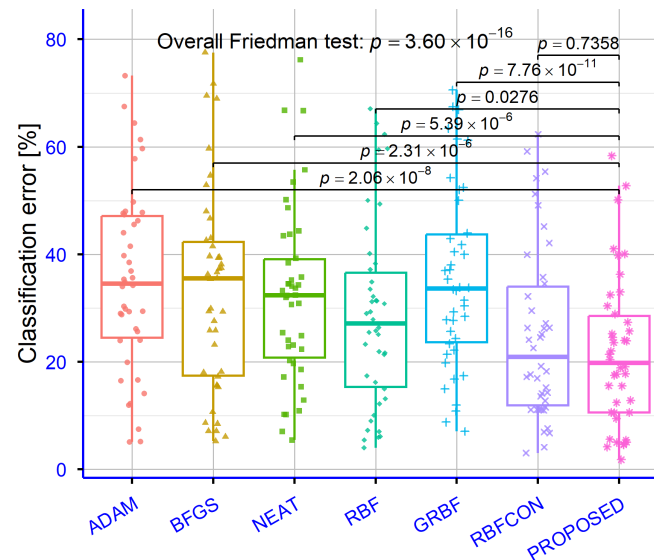


Figure 3. Classification error distribution across the six evaluated methods (overall Friedman test: $p = 3.60 \times 10^{-16}$).

Table 3. Experimental results on the used regression datasets using the series of machine learning methods.

Dataset	ADAM	BFGS	NEAT	RBF	GRBF	RBFCON	Proposed
ABALONE	4.30	5.69	9.88	7.37	9.98	6.14	6.37
AIRFOIL	0.005	0.003	0.067	0.27	0.121	0.004	0.004
AUTO	70.84	60.97	56.06	17.87	16.78	11.03	10.02
BK	0.025	0.28	0.15	0.02	0.023	0.02	0.021
BL	0.622	2.55	0.05	0.013	0.005	0.04	0.0005
BASEBALL	77.90	119.63	100.39	93.02	98.91	66.74	53.20
CONCRETE	0.078	0.066	0.081	0.011	0.015	0.012	0.008
DEE	0.63	2.36	1.512	0.17	0.25	0.23	0.16
FA	0.048	0.426	0.19	0.015	0.15	0.013	0.019
FY	0.038	0.22	0.08	0.011	0.041	0.017	0.051
HO	0.035	0.62	0.169	0.03	0.076	0.013	0.007
HOUSING	80.998	97.38	56.49	57.68	95.69	20.6	10.36
LASER	0.03	0.015	0.084	0.03	0.075	0.06	0.0198
LW	0.028	2.98	0.17	0.03	0.057	0.011	0.016
MB	0.06	0.129	0.061	5.43	0.159	0.055	0.066
MORTGAGE	9.24	8.23	14.11	1.45	1.92	0.165	0.0096
NT	0.006	0.129	0.027	13.97	0.01	0.006	0.006
PLASTIC	11.71	20.32	20.77	8.62	25.91	3.58	2.30
PY	0.321	0.578	0.075	0.012	0.029	0.003	0.015
PL	0.117	0.290	0.098	2.118	0.155	0.064	0.024
QUAKE	0.039	0.012	0.102	0.01	0.79	0.036	0.036
SN	0.026	0.400	0.174	0.027	0.027	0.025	0.024
STOCK	180.893	302.43	215.818	12.23	25.18	8.31	1.208
TREASURY	11.16	9.91	15.52	2.02	1.89	0.003	0.047
VE	0.359	1.92	0.045	0.024	0.023	0.04	0.04
Average	17.98	25.50	19.69	8.90	11.13	4.69	3.36

Mean classification error for the incorporated classification datasets and average Friedman rank is outlined in Table 4.

The raw average regression error is disproportionately influenced by datasets with large target-value magnitudes (e.g., STOCK, BASEBALL); the scale-free average rank in Table 5 confirms the same overall ordering, with the Proposed method achieving the best (lowest) average rank of 2.06 across the 25 datasets.

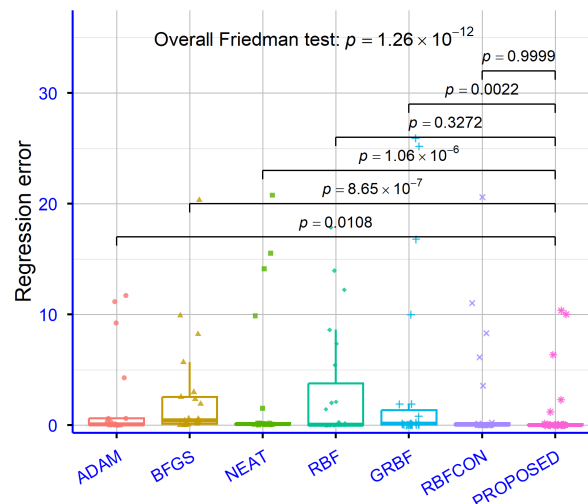


Figure 4. Regression error distribution for all evaluated methods (overall Friedman test: $p = 1.26 \times 10^{-12}$).

Table 4. Mean classification error and average Friedman rank across the 42 datasets (lower rank = better). Overall Friedman $\chi^2 = 84.82$, $p = 3.60 \times 10^{-16}$.

Method	Mean Error	Avg Rank
PROPOSED	21.58	2.02
RBFCN	24.96	2.74
RBF	28.68	3.61
NEAT	32.16	4.55
BFGS	33.25	4.64
ADAM	35.18	5.02
GRBF	35.50	5.42

Table 5. Mean regression error (raw scale) and average Friedman rank across the 25 datasets. The average rank is scale-free and is not dominated by high-magnitude datasets (e.g., STOCK, BASEBALL), unlike the raw mean error. Overall Friedman $\chi^2 = 67.61$, $p = 1.26 \times 10^{-12}$.

Method	Mean Error	Avg Rank
PROPOSED	3.36	2.06
RBFCN	4.69	2.24
RBF	8.90	3.58
ADAM	17.98	4.34
GRBF	11.13	4.62
NEAT	19.69	5.56
BFGS	25.50	5.60

Post hoc pairwise comparisons confirm that the proposed method differs significantly from ADAM ($p_{adj} = 0.0108$), GRBF ($p_{adj} = 0.0022$), NEAT ($p_{adj} = 1.06 \times 10^{-6}$), and BFGS ($p_{adj} = 8.65 \times 10^{-7}$) (all $p_{adj} < 0.05$), but the differences from RBFCN ($p_{adj} = 0.9999$) and RBF ($p_{adj} = 0.3272$) did not reach statistical significance, indicating that its advantage over these two closest competitors is directionally consistent but not conclusively established at the individual-comparison level.

For regression tasks (Table 3), the mean error values were 17.98 for ADAM, 25.50 for BFGS, 19.69 for NEAT, 8.90 for RBF, 11.13 for GRBF, 4.69 for RBFCN, and 3.36 for the proposed method. The overall Friedman test (Figure 4) yielded $p = 1.26 \times 10^{-12}$, again confirming statistically significant differences across methods. The proposed method achieves a mean error of 3.36, representing an improvement of about 28% over RBFCN

(4.69), which is a nearly threefold improvement over RBF (8.90), and nearly eightfold (not tenfold) improvement over BFGS (25.50).

3.4. Experiments with the Scale Parameter F

Table 6 and Figure 5 present the effect of the scale parameter F on classification performance, while Table 7 and Figure 6 present the corresponding results for regression tasks. For classification (Table 6), the mean error rates were 25.93% for $F = 1$, 22.85% for $F = 2$, 21.58% for $F = 4$, and 21.24% for $F = 8$. The overall Friedman test (Figure 5) yielded $p = 6.43 \times 10^{-5}$, confirming that the choice of F has a statistically significant impact on performance. A clear decreasing trend in error is observed as F increases, with $F = 8$ yielding a marginally lower average than $F = 4$. However, the difference between these two settings is minimal (21.58% versus 21.24%), suggesting that $F = 4$ represents a practical default with a favorable performance-to-complexity trade-off. For regression (Table 7), the mean error values were 4.69 for $F = 1$, 4.03 for $F = 2$, 3.36 for $F = 4$, and 3.29 for $F = 8$, with the Friedman test (Figure 6) yielding $p = 3.09 \times 10^{-4}$. A consistent improvement is again observed with increasing F , and the results for $F = 4$ and $F = 8$ are closely comparable across regression datasets.

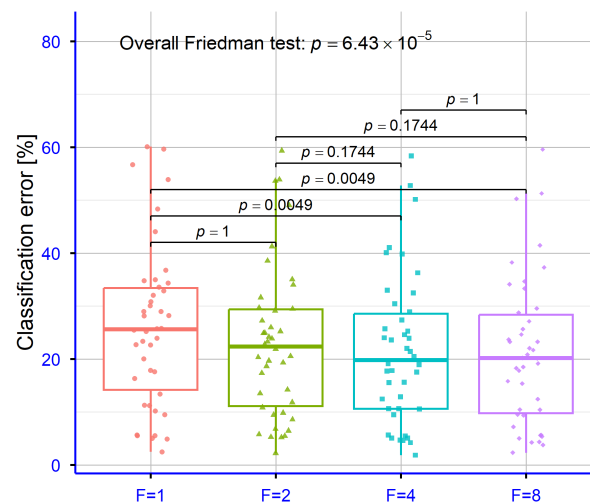


Figure 5. Effect of the scale parameter F on classification accuracy of the proposed method (overall Friedman test: $p = 6.43 \times 10^{-5}$).

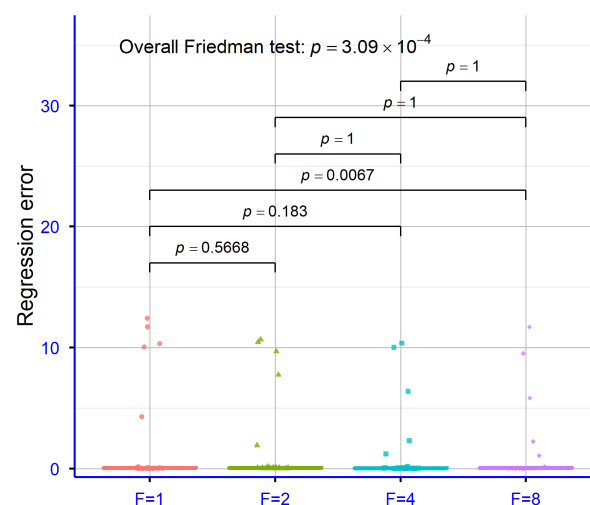


Figure 6. Sensitivity of regression performance to the scale parameter F (overall Friedman test: $p = 3.09 \times 10^{-4}$).

Table 6. Experimental results on the classification datasets, using the proposed method and different values for scale parameter F .

Dataset	$F = 1$	$F = 2$	$F = 4$	$F = 8$
APPENDICITIS	17.60%	14.20%	17.80%	18.50%
ALCOHOL	30.09%	23.83%	20.43%	20.21%
AUSTRALIAN	16.35%	18.64%	18.90%	20.13%
BALANCE	34.97%	23.28%	17.50%	18.27%
BANDS	34.33%	35.03%	36.28%	34.14%
CIRCULAR	34.83%	8.55%	5.04%	3.76%
CLEVELAND	48.38%	49.03%	50.14%	50.28%
DERMATOLOGY	60.10%	38.57%	39.80%	38.26%
ECOLI	56.73%	59.36%	58.36%	59.61%
FERT	25.80%	22.80%	21.40%	21.70%
GLASS	53.90%	53.65%	52.76%	51.29%
HABERMAN	28.23%	27.24%	27.37%	27.13%
HAYES ROTH	59.69%	53.92%	40.08%	37.31%
HEART	22.67%	24.15%	22.00%	20.85%
HEARTATTACK	20.03%	19.67%	19.13%	19.17%
HOUSEVOTES	5.52%	5.26%	4.61%	5.00%
IONOSPHERE	11.20%	13.51%	12.83%	12.43%
LIVERDISORDER	30.85%	29.65%	30.41%	29.56%
LYMOGRAPHY	32.86%	20.50%	24.57%	23.26%
MAMMOGRAPHIC	28.14%	21.88%	17.72%	17.80%
PARKINSONS	17.84%	17.32%	15.53%	15.37%
PAGE BLOCKS	9.48%	9.43%	9.47%	9.36%
PHONEME	23.37%	20.32%	15.59%	15.82%
PIMA	23.91%	24.87%	23.59%	23.71%
POPFAILURES	10.18%	6.43%	5.61%	5.63%
REGIONS2	25.73%	25.97%	25.71%	25.65%
RING	5.55%	5.45%	5.05%	5.57%
SAHEART	29.00%	29.13%	28.89%	28.78%
SATIMAGE	33.62%	34.01%	32.97%	34.68%
SEGMENT	32.05%	31.62%	32.47%	33.33%
SONAR	44.05%	29.45%	24.00%	22.05%
SPAMBASE	13.39%	10.84%	10.54%	10.44%
SPIRAL	36.80%	41.27%	41.03%	41.49%
STATHEART	22.74%	25.22%	23.89%	23.26%
STUDENT	25.50%	19.33%	12.43%	4.25%
TRANSFUSION	25.19%	24.92%	25.23%	24.65%
WDBC	4.86%	5.19%	4.19%	4.37%
WINE	11.29%	9.82%	10.59%	10.41%
Z_F_S	5.03%	6.80%	4.68%	5.37%
ZO_NF_S	5.67%	5.74%	5.47%	7.22%
ZONF_S	2.45%	2.22%	1.82%	2.30%
ZOO	29.00%	11.80%	10.60%	9.50%
Average	25.93%	22.85%	21.58%	21.24%

Table 7. Experimental results on the used regression datasets using the proposed method and a series of values for the parameter F .

Dataset	$F = 1$	$F = 2$	$F = 4$	$F = 8$
ABALONE	10.32	7.76	6.37	5.83
AIRFOIL	0.004	0.004	0.004	0.004
AUTO	10.06	9.68	10.02	9.53
BK	0.025	0.02	0.021	0.003

Table 7. Cont.

Dataset	$F = 1$	$F = 2$	$F = 4$	$F = 8$
BL	0.0098	0.0052	0.0005	0.041
BASEBALL	67.54	59.65	53.20	51.28
CONCRETE	0.027	0.022	0.008	0.005
DEE	0.16	0.16	0.16	0.16
FA	0.02	0.019	0.019	0.017
FY	0.046	0.047	0.051	0.053
HO	0.007	0.007	0.007	0.007
HOUSING	12.43	10.45	10.36	11.70
LASER	0.096	0.044	0.0198	0.009
LW	0.032	0.028	0.016	0.014
MB	0.067	0.066	0.066	0.06
MORTGAGE	0.011	0.009	0.0096	0.011
NT	0.006	0.006	0.006	0.006
PLASTIC	11.74	10.66	2.30	2.24
PY	0.029	0.017	0.015	0.014
PL	0.117	0.106	0.024	0.022
QUAKE	0.036	0.036	0.036	0.036
SN	0.031	0.025	0.024	0.023
STOCK	4.28	1.92	1.208	1.08
TREASURY	0.041	0.044	0.047	0.052
VE	0.026	0.034	0.04	0.035
Average	4.69	4.03	3.36	3.29

3.5. Experiments with Number of Chromosomes N_c

Table 8 and Figure 7 illustrate the effect of the chromosome population size N_c on classification performance, while Table 9 and Figure 8 present the corresponding regression results. For classification (Table 8), the mean error rates were 23.03% for $N_c = 100$, 22.44% for $N_c = 200$, 21.58% for $N_c = 500$, and 21.55% for $N_c = 1000$. The overall Friedman test (Figure 7) yielded $p = 0.0010$, confirming a statistically significant effect of N_c on classification accuracy. A steady improvement is observed from $N_c = 100$ to $N_c = 500$, while the increase to $N_c = 1000$ yields no meaningful additional gain, suggesting that a population of 500 chromosomes offers a well-balanced trade-off between computational cost and predictive performance. For regression (Table 9), the mean error values were 3.63 for $N_c = 100$, 3.45 for $N_c = 200$, 3.36 for $N_c = 500$, and 3.55 for $N_c = 1000$. The overall Friedman test (Figure 8) yielded $p = 0.2415$, a value that does not reach statistical significance, indicating that for regression tasks the chromosome count does not substantially influence performance and that $N_c = 500$ remains a sufficient and well-justified setting.

Furthermore, the proposed method was compared to the GRBF method on the classification datasets for a range of population sizes (i.e., different numbers of chromosomes). The comparison was performed with respect to both the average classification error and the execution time. The comparison for the average classification error is presented in Figure 9 and the comparison for the execution time is depicted in Figure 10.

The proposed method significantly outperforms the GRBF technique in terms of average classification error. Moreover, the classification error remains essentially unchanged for population sizes exceeding 500 chromosomes, supporting the choice of this value for the parameter N_c . However, the execution time of the proposed method is considerably higher than that of the GRBF method, particularly for large population sizes. This increase is expected and can be attributed to the greater computational complexity of the proposed approach compared to the GRBF method.

Table 8. Experimental results on the classification datasets, using the proposed method and different values the number of chromosomes denoted as N_c .

Dataset	$N_c = 100$	$N_c = 200$	$N_c = 500$	$N_c = 1000$
APPENDICITIS	20.00%	19.40%	17.80%	15.40%
ALCOHOL	25.32%	20.21%	20.43%	19.36%
AUSTRALIAN	23.87%	20.94%	18.90%	17.81%
BALANCE	17.18%	16.97%	17.50%	17.84%
BANDS	38.28%	36.61%	36.28%	36.17%
CIRCULAR	11.10%	6.42%	5.04%	5.29%
CLEVELAND	48.66%	48.86%	50.14%	50.38%
DERMATOLOGY	41.69%	41.94%	39.80%	40.83%
ECOLI	60.40%	60.76%	58.36%	59.52%
FERT	22.50%	21.10%	21.40%	24.00%
GLASS	55.99%	55.14%	52.76%	51.62%
HABERMAN	26.17%	26.77%	27.37%	27.53%
HAYES ROTH	41.61%	42.39%	40.08%	38.08%
HEART	22.85%	23.34%	22.00%	21.59%
HEARTATTACK	17.90%	18.87%	19.13%	18.83%
HOUSEVOTES	6.65%	6.35%	4.61%	5.35%
IONOSPHERE	14.66%	14.12%	12.83%	11.83%
LIVERDISORDER	29.50%	29.59%	30.41%	29.77%
LYMOGRAPHY	29.72%	29.79%	24.57%	22.43%
MAMMOGRAPHIC	18.27%	17.77%	17.72%	17.84%
PARKINSONS	15.95%	15.68%	15.53%	16.05%
PAGE BLOCKS	9.40%	9.36%	9.47%	10.43%
PHONEME	16.00%	15.54%	15.59%	16.37%
PIMA	23.50%	23.66%	23.59%	23.23%
POPFAILURES	5.91%	6.33%	5.61%	5.46%
REGIONS2	25.71%	25.66%	25.71%	25.43%
RING	5.88%	5.88%	5.05%	4.46%
SAHEART	28.46%	28.00%	28.89%	28.94%
SATIMAGE	34.96%	35.21%	32.97%	35.15%
SEGMENT	33.05%	32.73%	32.47%	31.95%
SONAR	25.75%	25.00%	24.00%	22.95%
SPAMBASE	12.41%	11.74%	10.54%	10.11%
SPIRAL	40.59%	41.39%	41.03%	41.11%
STATHEART	23.78%	24.30%	23.89%	25.11%
STUDENT	22.18%	17.10%	12.43%	11.45%
TRANSFUSION	25.91%	25.74%	25.23%	25.34%
WDBC	4.54%	4.54%	4.19%	4.75%
WINE	11.00%	10.12%	10.59%	10.71%
Z_F_S	5.90%	5.43%	4.68%	6.07%
ZO_NF_S	7.74%	7.34%	5.47%	5.70%
ZONF_S	3.26%	2.52%	1.82%	2.60%
ZOO	13.10%	11.90%	10.60%	10.40%
Average	23.03%	22.44%	21.58%	21.55%

Table 9. Experimental results on the used regression datasets using the proposed method and different values for the number of chromosomes N_c .

Dataset	$N_c = 100$	$N_c = 200$	$N_c = 500$	$N_c = 1000$
ABALONE	7.24	6.90	6.37	6.13
AIRFOIL	0.004	0.004	0.004	0.004
AUTO	9.70	9.57	10.02	9.88
BK	0.02	0.02	0.021	0.022

Table 9. Cont.

Dataset	$N_c = 100$	$N_c = 200$	$N_c = 500$	$N_c = 1000$
BL	0.0007	0.0006	0.0005	0.022
BASEBALL	56.15	53.69	53.20	56.45
CONCRETE	0.01	0.009	0.008	0.007
DEE	0.23	0.16	0.16	0.16
FA	0.025	0.022	0.019	0.016
FY	0.049	0.049	0.051	0.051
HO	0.007	0.007	0.007	0.008
HOUSING	10.50	10.92	10.36	11.94
LASER	0.058	0.025	0.0198	0.017
LW	0.02	0.018	0.016	0.014
MB	0.06	0.063	0.066	0.065
MORTGAGE	0.014	0.008	0.0096	0.01
NT	0.006	0.006	0.006	0.006
PLASTIC	5.11	3.37	2.30	2.32
PY	0.015	0.016	0.015	0.013
PL	0.045	0.032	0.024	0.023
QUAKE	0.036	0.036	0.036	0.036
SN	0.023	0.023	0.024	0.024
STOCK	1.31	1.22	1.208	1.38
TREASURY	0.048	0.048	0.047	0.047
VE	0.039	0.042	0.04	0.047
Average	3.63	3.45	3.36	3.55

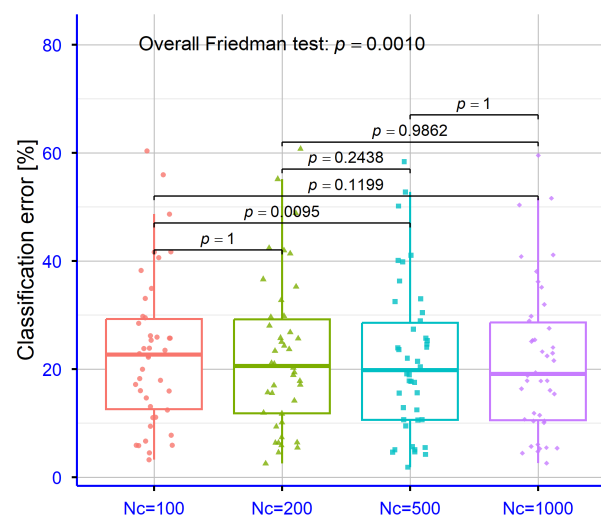


Figure 7. Impact of population size N_c on classification results of the proposed method (overall Friedman test: $p = 0.0010$).

3.6. Experiments with the Major Parts of the Method

An additional set of ablation experiments was conducted by removing key components of the proposed methodology to assess its robustness and quantify the contribution of each component to the overall performance. In the first experiment, the local optimization procedure was removed, and the experiments were carried out on both the classification and regression datasets for different values of the critical parameter F . The average classification error for all classification datasets is depicted in Figure 11. Similarly, Figure 12 presents the average regression error obtained by the application of the proposed method to the used regression datasets. The column LOCAL denoted the incorporation of the BFGS optimizer in the proposed method and the column NOLOCAL represents the application of the proposed method without the BFGS optimizer.

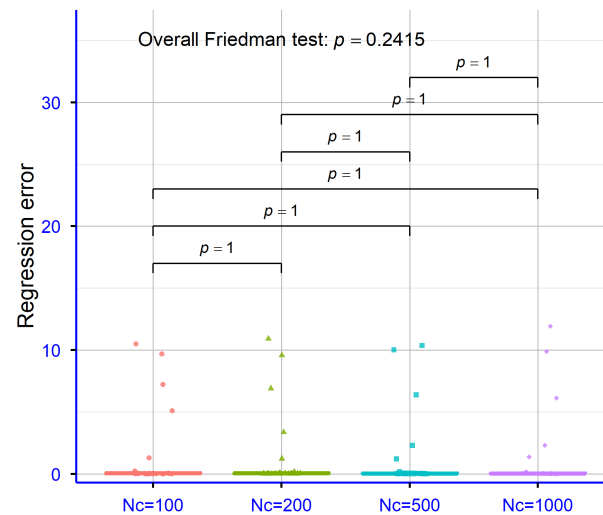


Figure 8. Regression error under varying chromosome population sizes N_c (overall Friedman test: $p = 0.2415$).

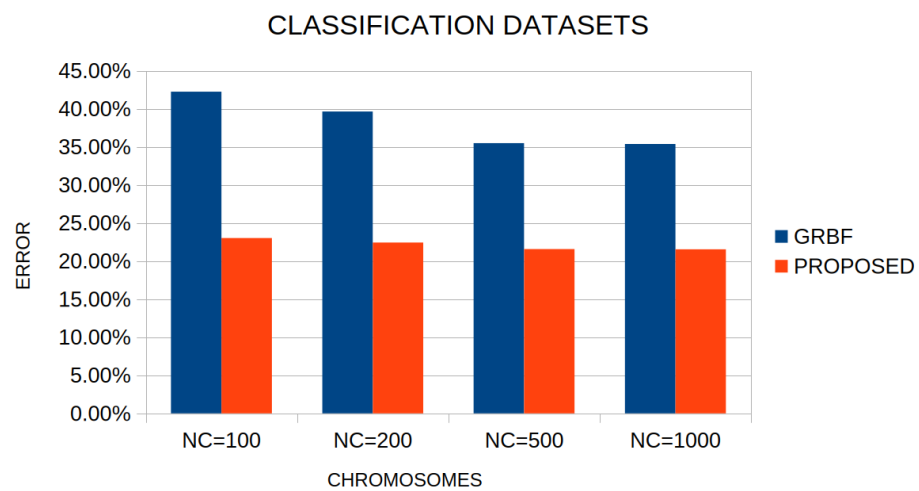


Figure 9. Comparison of average classification error between the proposed method and the GRBF method on the series of selected classification datasets for different values of used chromosomes.

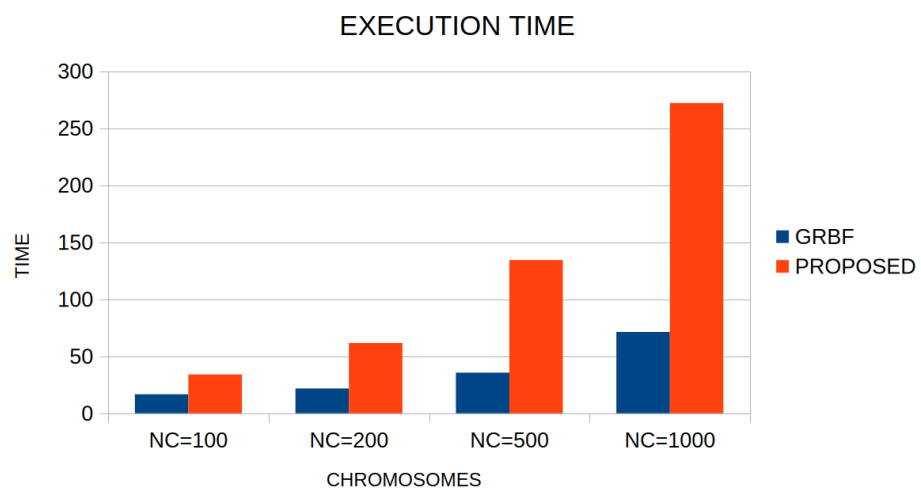


Figure 10. Comparison of average execution time between the proposed method and the GRBF method on the series of classification datasets, for different values of number of chromosomes.

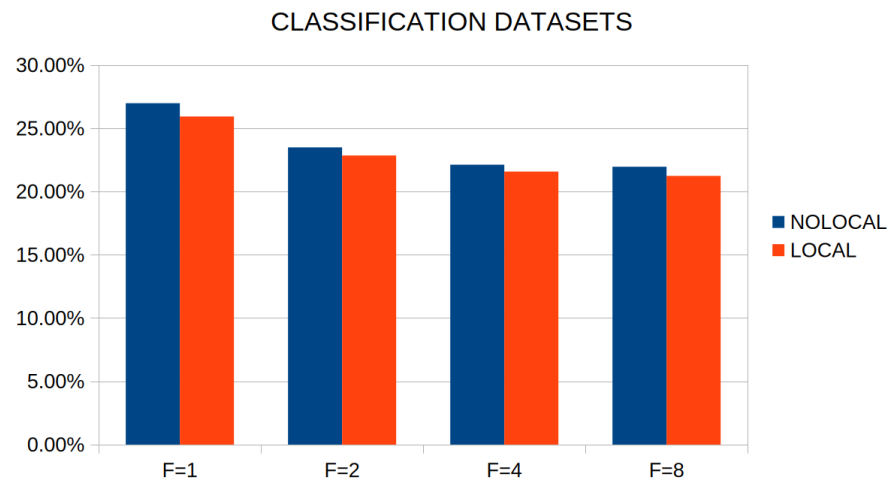


Figure 11. Average classification error of the proposed method for the series of classification datasets using different values for parameter F with and without local optimization.



Figure 12. Average regression error of the proposed method for the regression datasets using different values for parameter F with and without local optimization.

The above plots clearly demonstrate that the incorporation of the local optimization procedure improves the performance of the proposed technique, with a more pronounced improvement observed on the regression datasets than on the classification datasets.

In the next experiment, an attempt was made to evaluate the effectiveness of the k-means algorithm for estimating the initial range of values of the RBF network parameters. Figure 13 presents the average classification error using the proposed method and different initialization ranges and Figure 14 stands for the average regression error using the proposed method and the same initialization ranges. The columns presented in graphs have the following meaning:

1. The column INIT = 1 denotes the application of the proposed method, where the variables of the RBF network are initialized in range $[-1, 1]$.
2. The column INIT = 10 represents the application of the proposed method and initialization for the variables in range $[-10, 10]$.
3. The column INIT = 20 stands for the incorporation of the proposed method and initialization for the variables in range $[-20, 20]$.
4. The column PROPOSED represents the usage of the proposed method with the initialization presented in this article with the k-means technique.

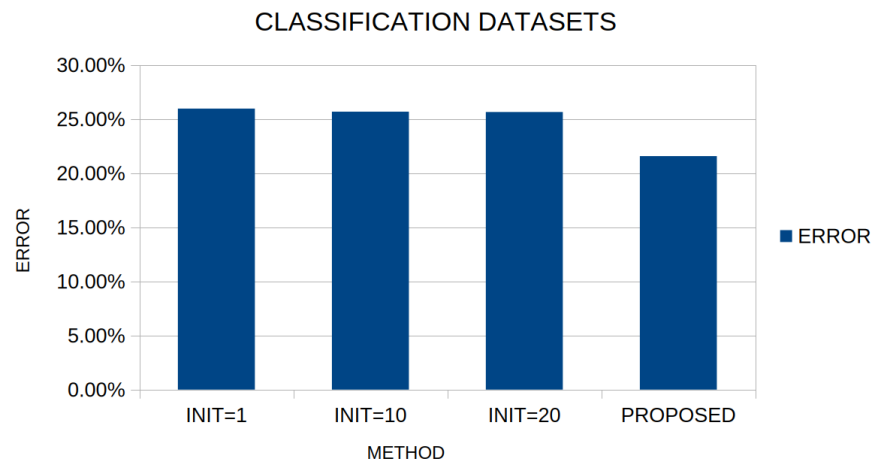


Figure 13. Average classification error obtained in the used classification datasets using the proposed method and different initialization ranges.

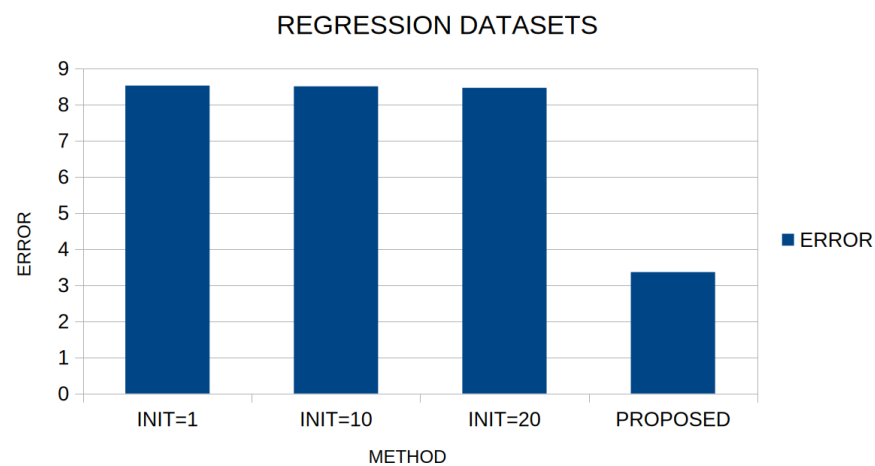


Figure 14. Average classification error obtained in the used regression datasets with the application of the proposed method and different initialization ranges.

It is evident that, in both cases (classification and regression), the proposed initialization technique significantly improves the performance of the proposed method.

4. Conclusions

This paper introduced a two-phase evolutionary methodology for the effective training of Radial Basis Function networks. In the first phase, the K-means clustering algorithm is employed to establish informed search bounds for the network centers and variances, thereby constraining the subsequent optimization within a meaningful parameter space. In the second phase, a genetic algorithm operates within these bounds to identify optimal center and variance configurations, while the network weights are determined analytically by solving a system of linear equations. The proposed method was evaluated across a large and diverse collection of 42 classification and 25 regression benchmark datasets, consistently outperforming five competing approaches, namely ADAM, BFGS, NEAT, RBF, and GRBF. The mean classification error of 21.58% and the mean regression error of 3.36 represent substantial improvements over all baselines, and the differences were confirmed to be statistically significant by the Friedman test. The sensitivity analysis further demonstrated that the method is robust across a range of parameter settings, with a population size of 500 chromosomes and a scale factor of $F = 4$, providing a reliable and computationally efficient default configuration.

While the proposed method achieved the lowest error on roughly half of the benchmark datasets (21 of 42 classification and 13 of 25 regression datasets), it was outperformed by at least one baseline on the remaining cases, and these cases follow identifiable patterns. For classification, RBFCON emerged as the most competitive alternative, achieving the lowest error on seven datasets (AUSTRALIAN, CLEVELAND, HAYES ROTH, HEART, HOUSEVOTES, PARKINSONS, and STATHEART), a cluster of predominantly medium-sized, and binary or near-binary medical and economic classification problems. This is consistent with the post hoc analysis (Table 4), which shows that the performance gap between the proposed method and RBFCON does not reach statistical significance ($p_{adj} = 0.7358$), unlike the gaps with all other baselines. Beyond RBFCON, the remaining losses largely follow the previously observed pattern of small sample size (e.g., APPENDICITIS, FERT, HABERMAN, STUDENT, POPFAILURES) or a relatively large number of classes with limited samples per class (e.g., DERMATOLOGY, ECOLI, GLASS, BANDS, BALANCE, TRANSFUSION), where a fixed number of hidden units ($k = 10$) and limited training data may not provide enough signal for the evolutionary search to consistently outperform simpler, directly-optimized alternatives. Notably, in several of these cases (e.g., MAMMOGRAPHIC, PAGE BLOCKS, PHONEME), the performance gap with the best-performing baseline was under one percentage point and is unlikely to be practically meaningful. For regression, RBFCON was again the strongest competing method, prevailing on FA, LW, MB, PY, and TREASURY, while RBF remained competitive on BK, FY, QUAKE, and VE; correspondingly, the post hoc tests (Table 5) show that the proposed method's advantage over both RBFCON ($p_{adj} = 0.9999$) and RBF ($p_{adj} = 0.3272$) does not reach statistical significance, whereas its advantage over ADAM, BFGS, NEAT, and GRBF is significant ($p_{adj} < 0.05$ in all four cases). These datasets are generally characterized by low target-value magnitude and low intrinsic noise, where traditional RBF training, grammar-guided architecture search, or direct gradient-based optimization already achieve a near-optimal fit; in such cases, the broader exploration performed by the genetic algorithm offers limited additional benefit and may introduce marginal variance. Taken together, these observations suggest that the advantage of the proposed evolutionary training is most pronounced on larger, higher-dimensional, or structurally complex datasets, while on small-sample, low-noise, or already well-fit problems, RBFCON and, to a lesser extent, conventional RBF training may already be near-sufficient.

A limitation of the present study is that only the mean error across the 30 independent runs was retained for each dataset–method combination; per-run standard deviations were not stored during the original experiments and could therefore not be reported. Future revisions of the experimental protocol will retain per-run results to support variability analysis alongside mean performance.

Several directions are proposed for future investigation. First, the computational cost of the genetic algorithm could be reduced through parallelization, enabling the simultaneous evaluation of multiple candidate solutions across multi-core or distributed computing environments. Second, adaptive strategies for the automatic selection of key parameters, such as the population size N_c , the number of generations N_g , and the scale factor F , could further reduce the need for manual tuning while improving performance across diverse problem types. Third, the current Gaussian basis function could be replaced or complemented by alternative kernel functions, such as multiquadric or inverse multiquadric kernels, potentially broadening the class of problems the model can address effectively. Fourth, the integration of feature selection mechanisms within the training pipeline could improve both accuracy and interpretability, particularly for high-dimensional datasets. Finally, extending the proposed framework to deep or multi-hidden-layer RBF architectures represents a promising avenue for tackling more complex learning tasks.

Author Contributions: V.C. and I.G.T. conducted the experiments, employing several datasets and provided the comparative experiments. D.T. and V.C. performed the statistical analysis and prepared the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been financed by the European Union: Next Generation EU through the Program Greece 2.0 National Recovery and Resilience Plan, under the call RESEARCH—CREATE—INNOVATE, project name “iCREW: Intelligent small craft simulator for advanced crew training using Virtual Reality techniques” (project code: TAEDK-06195).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Raissi, M.; Karniadakis, G.E. Hidden physics models: Machine learning of nonlinear partial differential equations. *J. Comput. Phys.* **2018**, *357*, 125–141.
2. Kashinath, K.; Mustafa, M.; Albert, A.; Wu, J.L.; Jiang, C.; Esmaeilzadeh, S.; Azizzadenesheli, K.; Wang, R.; Chattopadhyay, A.; Singh, A.; et al. Physics-informed machine learning: Case studies for weather and climate modelling. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200093.
3. Viqar, M.; Basak, S.; Dasgupta, A.; Agrawal, S.; Saha, S. Machine learning in astronomy: A case study in quasar-star classification. In *Proceedings of the 1st International Conference on Emerging Technologies in Data Mining and Information Security, Kolkata, India, 23–25 February 2018*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 3, pp. 827–836.
4. Luo, S.; Leung, A.P.; Hui, C.Y.; Li, K.L. An investigation on the factors affecting machine learning classifications in gamma-ray astronomy. *Mon. Not. R. Astron. Soc.* **2020**, *492*, 5377–5390.
5. Meuwly, M. Machine learning for chemical reactions. *Chem. Rev.* **2021**, *121*, 10218–10239.
6. Aguiar, J.A.; Gong, M.L.; Tasdizen, T. Crystallographic prediction from diffraction and chemistry data for higher throughput classification using machine learning. *Comput. Mater. Sci.* **2020**, *173*, 109409.
7. Yadav, S.S.; Jadhav, S.M. Deep convolutional neural network based medical image classification for disease diagnosis. *J. Big Data* **2019**, *6*, 113.
8. Qing, L.; Linhong, W.; Xuehai, D. A Novel Neural Network-Based Method for Medical Text Classification. *Future Internet* **2019**, *11*, 255.
9. Athey, S. The impact of machine learning on economics. In *The Economics of Artificial Intelligence: An Agenda*; University of Chicago Press: Chicago, IL, USA, 2018; pp. 507–547.
10. Hafezi, R.; Shahrabadi, J.; Hadavandi, E. A bat-neural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price. *Appl. Soft Comput.* **2015**, *29*, 196–210.
11. Ghai, D.; Tripathi, S.L.; Saxena, S.; Chanda, M.; Alazab, M. (Eds.) *Machine Learning Algorithms for Signal and Image Processing*; John Wiley & Sons: Hoboken, NJ, USA, 2022.
12. Ahmed, N.K.; Atiya, A.F.; Gayar, N.E.; El-Shishiny, H. An empirical comparison of machine learning models for time series forecasting. *Econom. Rev.* **2010**, *29*, 594–621.
13. Huang, Z.; Chen, H.; Hsu, C.-J.; Chen, W.-H.; Wu, S. Credit rating analysis with support vector machines and neural networks: A market comparative study. *Decis. Support Syst.* **2004**, *37*, 543–558.
14. Peta, K.; Żurek, J. Prediction of air leakage in heat exchangers for automotive applications using artificial neural networks. In *Proceedings of the 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, 8–10 November 2018; pp. 721–725.
15. Montazer, G.A.; Giveki, D.; Karami, M.; Rastegar, H. Radial basis function neural networks: A review. *Comput. Rev. J.* **2018**, *1*, 52–74.
16. Teng, P. Machine-learning quantum mechanics: Solving quantum mechanics problems using radial basis function networks. *Phys. Rev. E* **2018**, *98*, 033305.
17. Jovanović, R.; Sretenovic, A. Ensemble of radial basis neural networks with K-means clustering for heating energy consumption prediction. *FME Trans.* **2017**, *45*, 51–57.
18. Gorbachenko, V.I.; Zhukov, M.V. Solving boundary value problems of mathematical physics using radial basis function networks. *Comput. Math. Math. Phys.* **2017**, *57*, 145–155.

19. Määttä, J.; Bazaliy, V.; Kimari, J.; Djurabekova, F.; Nordlund, K.; Roos, T. Gradient-based training and pruning of radial basis function networks with an application in materials physics. *Neural Netw.* **2021**, *133*, 123–131.
20. Bhatia, G.S.; Arora, G. Radial basis function methods for solving partial differential equations—A review. *Indian J. Sci. Technol.* **2016**, *9*, 1–18.
21. Cavoretto, R.; De Rossi, A. A two-stage adaptive scheme based on RBF collocation for solving elliptic PDEs. *Comput. Math. with Appl.* **2020**, *79*, 3206–3222.
22. Feng, R.; Duan, J. High accurate finite differences based on RBF interpolation and its application in solving differential equations. *J. Sci. Comput.* **2018**, *76*, 1785–1812.
23. Song, Q.; Li, S.; Bai, Q.; Yang, J.; Zhang, A.; Zhang, X.; Zhe, L. Trajectory planning of robot manipulator based on RBF neural network. *Entropy* **2021**, *23*, 1207.
24. Liu, Q.; Li, D.; Ge, S.S.; Ji, R.; Ouyang, Z.; Tee, K.P. Adaptive bias RBF neural network control for a robotic manipulator. *Neurocomputing* **2021**, *447*, 213–223.
25. Roh, S.B.; Oh, S.K.; Yoon, J.H.; Seo, K. Design of face recognition system based on fuzzy transform and radial basis function neural networks. *Soft Comput.* **2019**, *23*, 4969–4985.
26. Lopez-Martin, M.; Sanchez-Esguevillas, A.; Arribas, J.I.; Carro, B. Network intrusion detection based on extended RBF neural network with offline reinforcement learning. *IEEE Access* **2021**, *9*, 153153–153170.
27. Chen, Z. Research on internet security situation awareness prediction technology based on improved RBF neural network algorithm. *J. Comput. Cogn. Eng.* **2022**, *1*, 103–108.
28. Tao, J.; Yu, Z.; Zhang, R.; Gao, F. RBF neural network modeling approach using PCA based LM-GA optimization for coke furnace system. *Appl. Soft Comput.* **2021**, *111*, 107691.
29. Kanti, P.; Sharma, K.V.; Yashawantha, K.M.; Jamei, M.; Said, Z. Properties of water-based fly ash-copper hybrid nanofluid for solar energy applications: Application of RBF model. *Sol. Energy Mater. Sol. Cells* **2022**, *234*, 111423.
30. Yu, Y. GDP economic forecasting model based on improved RBF neural network. *Math. Probl. Eng.* **2022**, *2022*, 7630268.
31. Zhang, Q.; Abdullah, A.R.; Chong, C.W.; Ali, M.H. A Study on Regional GDP Forecasting Analysis Based on Radial Basis Function Neural Network with Genetic Algorithm (RBFNN-GA) for Shandong Economy. *Comput. Intell. Neurosci.* **2022**, *2022*, 8235308.
32. Sohrabi, P.; Jodeiri Shokri, B.; Dehghani, H. Predicting coal price using time series methods and combination of radial basis function (RBF) neural network with time series. *Min. Econ.* **2021**, *36*, 207–216.
33. Anyanwu, G.O.; Nwakanma, C.I.; Lee, J.M.; Kim, D.S. Optimization of RBF-SVM kernel using grid search algorithm for DDoS attack detection in SDN-based VANET. *IEEE Internet Things J.* **2022**, *10*, 8477–8490.
34. Gan, M.; Peng, H.; Dong, X.P. A hybrid algorithm to optimize RBF network architecture and parameters for nonlinear time series prediction. *Appl. Math. Model.* **2012**, *36*, 2911–2919.
35. Sideratos, G.; Hatziaargyriou, N. Using Radial Basis Neural Networks to Estimate Wind Power Production. In Proceedings of the 2007 IEEE Power Engineering Society General Meeting, Tampa, FL, USA, 24–28 June 2007; pp. 1–7. <https://doi.org/10.1109/PES.2007.385812>.
36. Regis, R.G. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Eng. Optim.* **2014**, *46*, 218–243.
37. Wu, Y.; Yin, Q.; Jie, H.; Wang, B.; Zhao, J. A RBF-based constrained global optimization algorithm for problems with computationally expensive objective and constraints. *Struct. Multidiscip. Optim.* **2018**, *58*, 1633–1655.
38. Kuncheva, L.I. Initializing of an RBF Network by a Genetic Algorithm. *Neurocomputing* **1997**, *14*, 273–288.
39. Ros, F.; Pintore, M.; Deman, A.; Chrétien, J.R. Automatical initialization of RBF neural networks. *Chemom. Intell. Lab. Syst.* **2007**, *87*, 26–32.
40. Wang, D.; Zeng, X.J.; Keane, J.A. A Clustering Algorithm for Radial Basis Function Neural Network Initialization. *Neurocomputing* **2012**, *77*, 144–155.
41. Benoudjit, N.; Verleysen, M. On the Kernel Widths in Radial-Basis Function Networks. *Neural Process. Lett.* **2003**, *18*, 139–154.
42. Neruda, R.; Kudova, P. Learning methods for radial basis function networks. *Future Gener. Comput. Syst.* **2005**, *21*, 1131–1142.
43. Ricci, E.; Perfetti, R. Improved pruning strategy for radial basis function networks with dynamic decay adjustment. *Neurocomputing* **2006**, *69*, 1728–1732.
44. Huang, G.-B.; Saratchandran, P.; Sundararajan, N. A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation. *IEEE Trans. Neural Netw.* **2005**, *16*, 57–67.
45. Bortman, M.; Aladjem, M. A Growing and Pruning Method for Radial Basis Function Networks. *IEEE Trans. Neural Netw.* **2009**, *20*, 1039–1045.
46. Yokota, R.; Barba, L.A.; Knepley, M.G. PetRBF—A parallel O(N) algorithm for radial basis function interpolation with Gaussians. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 1793–1804.
47. Lu, C.; Ma, N.; Wang, Z. Fault detection for hydraulic pump based on chaotic parallel RBF network. *EURASIP J. Adv. Signal Process.* **2011**, *49*, 2011.

48. Park, J.; Sandberg, I.W. Universal Approximation Using Radial-Basis-Function Networks. *Neural Comput.* **1991**, *3*, 246–257. <https://doi.org/10.1162/neco.1991.3.2.246>.
49. Mohammadi, R.; Ghomi, S.F.; Zeinali, F. A new hybrid evolutionary based RBF networks method for forecasting time series: A case study of forecasting emergency supply demand time series. *Eng. Appl. Artif. Intell.* **2014**, *36*, 204–214.
50. Wu, J.; Long, J.; Liu, M. Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm. *Neurocomputing* **2015**, *148*, 136–142.
51. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 153–165.
52. Elansari, T.; Ouanan, M.; Bourray, H. Mixed radial basis function neural network training using genetic algorithm. *Neural Process. Lett.* **2023**, *55*, 10569–10587.
53. Mukherjee, A.; Gupta, D.; Bhattacharyya, D. Mass-Constrained hybrid Gaussian radial basis neural networks: Development, training, and applications to modeling nonlinear dynamic noisy chemical processes. *Comput. Chem. Eng.* **2025**, *197*, 109080.
54. Bajer, D.; Zorić, B.; Martinović, G. Automatic design of radial basis function networks through enhanced differential evolution. In *Hybrid Artificial Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 244–256.
55. Alexandridis, A.; Chondrodima, E.; Sarimveis, H. Cooperative learning for radial basis function networks using particle swarm optimization. *Appl. Soft Comput.* **2016**, *49*, 485–497.
56. Rani, H.J.; Victoire, T.A.A. Training radial basis function networks for wind speed prediction using PSO enhanced differential search optimizer. *PLoS ONE* **2018**, *13*, e0196871.
57. Moscato, P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurr. Comput. Program C3p Rep.* **1989**, *826*, 37.
58. Ong, Y.S.; Keane, A.J. Meta-Lamarckian learning in memetic algorithms. *IEEE Trans. Evol. Comput.* **2004**, *8*, 99–110.
59. Ali, S.; Kurz, J.A.; Oughton, S. A Novel RBF Neural Network-based Hybrid Technique and Its Applications. *Res. Sq.* 2025, preprint. <https://doi.org/10.21203/rs.3.rs-7088954/v1>.
60. Oti, E.U.; Olusola, M.O.; Eze, F.C.; Enogwe, S.U. Comprehensive Review of K-Means Clustering Algorithms. *Criterion* **2021**, *12*, 22–23.
61. Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley Publishing Company: Reading, MA, USA, 1989.
62. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Springer: Berlin, Germany, 1996.
63. Grady, S.A.; Hussaini, M.Y.; Abdullah, M.M. Placement of wind turbines using genetic algorithms. *Renew. Energy* **2005**, *30*, 259–270.
64. Parvaze, S.; Kumar, R.; Khan, J.N.; Al-Ansari, N.; Parvaze, S.; Vishwakarma, D.K.; Elbeltagi, A.; Kuriqi, A. Optimization of water distribution systems using genetic algorithms: A review. *Arch. Comput. Methods Eng.* **2023**, *30*, 4209–4244.
65. Gordini, N. A genetic algorithm approach for SMEs bankruptcy prediction: Empirical evidence from Italy. *Expert Syst. Appl.* **2014**, *41*, 6433–6445.
66. Ding, S.; Su, C.; Yu, J. An optimizing BP neural network algorithm based on genetic algorithm. *Artif. Intell. Rev.* **2011**, *36*, 153–162.
67. Chatterjee, S.; Laudato, M.; Lynch, L.A. Genetic algorithms and their statistical applications: An introduction. *Comput. Stat. Data Anal.* **1996**, *22*, 633–651. [https://doi.org/10.1016/0167-9473\(96\)00011-4](https://doi.org/10.1016/0167-9473(96)00011-4).
68. Cavorotto, R.; De Rossi, A.; Lancellotti, S.; Mezzanotte, D. Genetic Algorithm-Based Shape Parameter Tuning for Radial Basis Function Interpolation. *Oper. Res. Forum* **2026**, *7*, 26. <https://doi.org/10.1007/s43069-026-00610-9>.
69. Kaelo, P.; Ali, M.M. Integrated crossover rules in real coded genetic algorithms. *Eur. J. Oper. Res.* **2007**, *176*, 60–76.
70. Powell, M.J.D. A Tolerant Algorithm for Linearly Constrained Optimization Calculations. *Math. Program.* **1989**, *45*, 547–566.
71. Kelly, M.; Longjohn, R.; Nottingham, K. The UCI Machine Learning Repository. Available online: <https://archive.ics.uci.edu> (accessed on 26 June 2026).
72. Alcalá-Fdez, J.; Fernandez, A.; Luengo, J.; Derrac, J.; García, S.; Sánchez, L.; Herrera, F. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *J. Mult.-Valued Log. Soft Comput.* **2011**, *17*, 255–287.
73. Weiss, S.M.; Kulikowski, C.A. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 1991.
74. Tzimourta, K.D.; Tsoulos, I.; Bilerio, I.T.; Tzallas, A.T.; Tsiouras, M.G.; Giannakeas, N. Direct Assessment of Alcohol Consumption in Mental State Using Brain Computer Interfaces and Grammatical Evolution. *Inventions* **2018**, *3*, 51.
75. Quinlan, J.R. Simplifying Decision Trees. *Int. J. Man.-Mach. Stud.* **1987**, *27*, 221–234.
76. Shultz, T.; Mareschal, D.; Schmidt, W. Modeling Cognitive Development on Balance Scale Phenomena. *Mach. Learn.* **1994**, *16*, 59–88.
77. Evans, B.; Fisher, D. Overcoming process delays with decision tree induction. *IEEE Expert* **1994**, *9*, 60–66.
78. Zhou, Z.H.; Jiang, Y. NeC4.5: Neural ensemble based C4.5. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 770–773.
79. Setiono, R.; Leow, W.K. FERNN: An Algorithm for Fast Extraction of Rules from Neural Networks. *Appl. Intell.* **2000**, *12*, 15–25.

80. Demiroz, G.; Govenir, H.A.; Ilter, N. Learning Differential Diagnosis of Eryhemato-Squamous Diseases using Voting Feature Intervals. *Artif. Intell. Med.* **1998**, *13*, 147–165.
81. Horton, P.; Nakai, K. A Probabilistic Classification System for Predicting the Cellular Localization Sites of Proteins. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **1996**, *4*, 109–115.
82. B. Hayes-Roth, B.; Hayes-Roth, F. Concept learning and the recognition and classification of exemplars. *J. Verbal Learn. Verbal Behav.* **1977**, *16*, 321–338.
83. Kononenko, I.; Šimec, E.; Robnik-Šikonja, M. Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF. *Appl. Intell.* **1997**, *7*, 39–55.
84. French, R.M.; Chater, N. Using noise to compute error surfaces in connectionist networks: A novel means of reducing catastrophic forgetting. *Neural Comput.* **2002**, *14*, 1755–1769.
85. Dy, J.G.; Brodley, C.E. Feature Selection for Unsupervised Learning. *J. Mach. Learn. Res.* **2004**, *5*, 845–889.
86. Perantonis, S.J.; Virvilis, V. Input Feature Extraction for Multilayered Perceptrons Using Supervised Principal Component Analysis. *Neural Process. Lett.* **1999**, *10*, 243–252.
87. Garcke, J.; Griebel, M. Classification with sparse grids using simplicial basis functions. *Intell. Data Anal.* **2002**, *6*, 483–502.
88. Cestnik, G.; Kononenko, I.; Bratko, I. Assistant-86: A Knowledge-Elicitation Tool for Sophisticated Users. In *Progress in Machine Learning*; Bratko, I., Lavrac, N., Eds.; Sigma Press: Wilmslow, UK, 1987; pp. 31–45.
89. Elter, M.; Schulz-Wendland, R.; Wittenberg, T. The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process. *Med. Phys.* **2007**, *34*, 4164–4172.
90. Little, M.A.; McSharry, P.E.; Hunter, E.J.; Spielman, J.; Ramig, L.O. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. *IEEE Trans. BioMed Eng.* **2009**, *56*, 1015. <https://doi.org/10.1109/TBME.2008.2005954>.
91. Smith, J.W.; Everhart, J.E.; Dickson, W.C.; Knowler, W.C.; Johannes, R.S. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Symposium on Computer Applications and Medical Care*; IEEE Computer Society Press: Los Alamitos, CA, USA, 1988; pp. 261–265.
92. Lucas, D.D.; Klein, R.; Tannahill, J.; Ivanova, D.; Brandon, S.; Domyancic, D.; Zhang, Y. Failure analysis of parameter-induced simulation crashes in climate models. *Geosci. Model Dev.* **2013**, *6*, 1157–1171.
93. Giannakeas, N.; Tsiouras, M.G.; Tzallas, A.T.; Kyriakidi, K.; Tsianou, Z.E.; Manousou, P.; Hall, A.; Karvounis, E.C.; Tsianos, V.; Tsianos, E. A clustering based method for collagen proportional area extraction in liver biopsy images. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, Milan, Italy, 25–29 August 2015*; pp. 3097–3100.
94. Hastie, T.; Tibshirani, R. Non-parametric logistic and proportional odds regression. *JRSS-C (Appl. Stat.)* **1987**, *36*, 260–276.
95. Dash, M.; Liu, H.; Scheuermann, P.; Tan, K.L. Fast hierarchical clustering and its validation. *Data Knowl. Eng.* **2003**, *44*, 109–138.
96. Gorman, R.P.; Sejnowski, T.J. Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets. *Neural Netw.* **1988**, *1*, 75–89.
97. Cranor, L.F.; LaMacchia, B.A. Spam! *Commun. ACM* **1998**, *41*, 74–83.
98. Cortez, P.; Gonçalves Silva, A.M. Using data mining to predict secondary school student performance. In *Proceedings of 5th Future Business Technology Conference (FUBUTEC 2008)*; EUROSIS-ETI: Ostend, Belgium, 2008; pp. 5–12.
99. Yeh, I.-C.; Yang, K.-J.; Ting, T.-M. Knowledge discovery on RFM model using Bernoulli sequence. *Expert Syst. Appl.* **2009**, *36*, 5866–5871.
100. Wolberg, W.H.; Mangasarian, O.L. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proc. Natl. Acad. Sci. USA* **1990**, *87*, 9193–9196.
101. Raymer, M.; Doom, T.E.; Kuhn, L.A.; Punch, W.F. Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm. *IEEE Trans. Syst. Man Cybern. Part B Cybern. Publ. IEEE Syst. Man Cybern. Soc.* **2003**, *33*, 802–813.
102. Zhong, P.; Fukushima, M. Regularized nonsmooth Newton method for multi-class support vector machines. *Optim. Methods Softw.* **2007**, *22*, 225–236.
103. Andrzejak, R.G.; Lehnertz, K.; Mormann, F.; Rieke, C.; David, P.; Elger, C.E. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Phys. Rev. E* **2001**, *64*, 1–8.
104. Koivisto, M.; Sood, K. Exact Bayesian Structure Discovery in Bayesian Networks. *J. Mach. Learn. Res.* **2004**, *5*, 549–573.
105. Nash, W.J.; Sellers, T.L.; Talbot, S.R.; Cawthor, A.J.; Ford, W.B. The Population Biology of Abalone (*Haliotis Species*) in Tasmania. I. Blacklip Abalone (*H. Rubra*) from the North Coast and Islands of Bass Strait. *Sea Fish. Div. Tech. Rep.* **1994**, *48*, 411.
106. Brooks, T.F.; Pope, D.S.; Marcolini, A.M. *Airfoil Self-Noise and Prediction*; Technical Report, NASA RP-1218; NASA: Washington, DC, USA, 1989.
107. Yeh, I.C. Modeling of strength of high performance concrete using artificial neural networks. *Cem. Concr. Res.* **1998**, *28*, 1797–1808.
108. Harrison, D.; Rubinfeld, D.L. Hedonic prices and the demand for clean air. *J. Environ. Econ. Manag.* **1978**, *5*, 81–102.

109. Simonoff, J.S. *Smoothing Methods in Statistics*; Springer: Berlin, Germany, 1996.
110. Mackowiak, P.A.; Wasserman, S.S.; Levine, M.M. A critical appraisal of 98.6 degrees f, the upper limit of the normal body temperature, and other legacies of Carl Reinhold August Wunderlich. *J. Amer. Med. Assoc.* **1992**, *268*, 1578–1580.
111. King, R.D.; Muggleton, S.; Lewis, R.; Sternberg, M.J.E. Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. Nat. Acad. Sci. USA* **1992**, *89*, 11322–11326.
112. Sikora, M.; Wrobel, L. Application of rule induction algorithms for analysis of data collected by seismic hazard monitoring systems in coal mines. *Arch. Min. Sci.* **2010**, *55*, 91–114.
113. Sanderson, C.; Curtin, R. Armadillo: A template-based C++ library for linear algebra. *J. Open Source Softw.* **2016**, *1*, 26.
114. Tsoulos, I.G.; Charilogis, V.; Kyrou, G.; Stavrou, V.N.; Tzallas, A. OPTIMUS: A Multidimensional Global Optimization Package. *J. Open Source Softw.* **2025**, *10*, 7584.
115. Hogan, R.J. Adept C++ Software Library: User Guide. Available online: https://www.met.reading.ac.uk/clouds/adept/adept_documentation.pdf (accessed on 26 June 2026).
116. Kingma, D.P.; Ba, J.L. ADAM: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
117. Xue, Y.; Tong, Y.; Neri, F. An ensemble of differential evolution and Adam for training feed-forward neural networks. *Inf. Sci.* **2022**, *608*, 453–471.
118. Yuan, Y.X. A modified BFGS algorithm for unconstrained optimization. *IMA J. Numer. Anal.* **1991**, *11*, 325–332.
119. Bishop, C. *Bishop, Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995.
120. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **1989**, *2*, 303–314.
121. Stanley, K.O.; Miikkulainen, R. Evolving Neural Networks through Augmenting Topologies. *Evol. Comput.* **2002**, *10*, 99–127.
122. Ding, S.; Xu, L.; Su, C. An optimizing method of RBF neural network based on genetic algorithm. *Neural Comput. Applic* **2012**, *21*, 333–336.
123. Tsoulos, I.G.; Varvaras, I.; Charilogis, V. RbfCon: Construct radial basis function neural networks with grammatical evolution. *Software* **2024**, *3*, 549–568.
124. Ryan, C.; O'Neill, M.; Collins, J.J. (Eds.) *Handbook of Grammatical Evolution*; Springer International Publishing: Cham, Switzerland, 2018; Volume 1.

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.