MDPI

Article

From Initialization to Convergence: A Three-Stage technique for Robust RBF Network Training

Ioannis G. Tsoulos[1,*], Vasileios Charilogis[2], Dimitrios Tsalikakis[3]

[1] Department of Informatics and Telecommunications, University of Ioannina, Greece;itsoulos@uoi.gr
[2] Department of Informatics and Telecommunications, University of Ioannina, Greece; v.charilog@uoi.gr
[3] Department of Engineering Informatics and Telecommunications, University of Western Macedonia, 50100 Kozani, Greece; dtsalikakis@uowm.gr
* Correspondence: itsoulos@uoi.gr

Abstract

Radial Basis Function (RBF) networks are well established machine learning tools used in a variety of classification and regression problems. A key component of these networks is their radial functions. These networks acquire adaptive capabilities through a two-stage training technique in most cases. In the first stage, the centers and variances for the radial functions are estimated, and in the second stage, through the solution of a linear system, the external weights for the radial functions are adjusted. However, in many cases this training technique has reduced performance either due to instability in arithmetic operations or due to trapping in local minima of the training error. In this paper, a three-stage method is proposed to address the above problems. In the first stage, an initial estimate of the intervals for the network parameter values is made, in the second stage, the network parameter values are adjusted within the intervals of the first phase, and finally in the third stage of the proposed technique, a local optimization method is used for the final adjustment of the network parameters. The proposed method was tested for its efficiency on a wide series of regression and classification datasets from the related bibliography with exceptional results.

Keywords: Machine learning; Neural networks; Genetic algorithms; Optimization

Received:
Revised:
Accepted:
Published:

Citation: Tsoulos, I.G.; Charilogis, V.; Tsalikakis D. From Initialization to Convergence: A Three-Stage technique for Robust RBF Network Training. *Journal Not Specified* **2025**, *1*, 0. https://doi.org/

1. Introduction

Many practical problems can be tackled by machine learning models, such as problems occurred in physics [1,2], astronomy [3,4], chemistry [5,6], medicine [7,8], economics [9,10], image processing [11], time series forecasting [12] etc. Among the most used machine learning tools one can detect the Radial Basis Function (RBF) networks, expressed using the following definition:

$$R(\overrightarrow{x}) = \sum_{i=1}^{k} w_i \phi(\| \overrightarrow{x} - \overrightarrow{c_i} \|) \quad (1)$$

The symbols appeared in this equation are defined as follows:

1. The vector $\overrightarrow{x}$ stands for the input pattern. The dimension of each pattern is denoted by d.
2. The constant k denotes number of weights of the network. The vector $\overrightarrow{w}$ denotes these weights.
3. The vectors $\overrightarrow{c_i}$, $i = 1, .., k$ denote the centers of the network.

4. The function $\phi(x)$ usually represents a Gaussian function having the following form:

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

A plot of the Gaussian function $c = 0,\ \sigma = 1$ is shown in Figure 1.

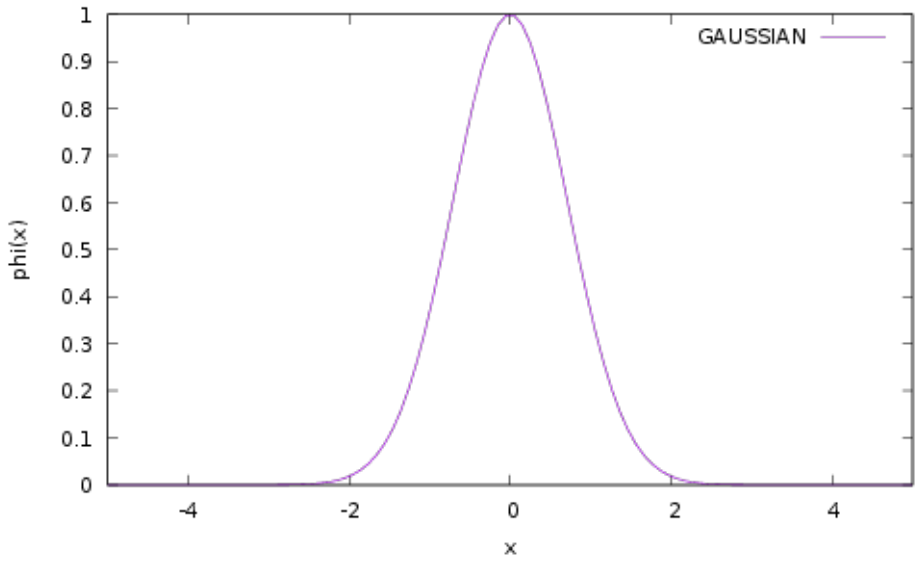

Figure 1. A typical plot for the Gaussian function.

This graph clearly shows that the value of the function decreases as the value of x moves away from the center c. The training error of any given $R(x)$ RBF network is defined as:

$$E\left(R\left(\overrightarrow{x}\right)\right) = \sum_{i=1}^{M}\left(R\left(\overrightarrow{x_i}\right) - y_i\right)^2 \tag{3}$$

The set $\left(\overrightarrow{x_i}, y_i\right),\ i = 1, ..., M$ denotes the training set of the objective problem and the values y_i are considered as the actual output for each pattern $\overrightarrow{x_i}$.

RBF networks have been used in many cases, such as face recognition [13], solutions of differential equations [14,15], stock prediction [16], robotics [17,18], network security [19,20], classification of process faults [21], time series prediction [22], estimation of wind power production [23] etc. Moreover, Park et al [24] proved that an RBF network with one processing layer is capable of universal approximation.

Recently, a series of papers have been proposed for the initialization of the parameters of RBF networks [25–27]. Moreover, Benoudjit et al provided a discussion on the estimation of kernel widths in RBF networks [28]. Additionally, a series of pruning techniques [29–31] have been introduced aiming to reduce the number of parameters of the RBF networks in order to avoid the overfitting problem. Also, a series of optimization techniques have been incorporated in the past to tackle the equation 3, such as Genetic algorithms [32,33], the Particle Swam Optimization method [34,35], the Differential Evolution technique [36] etc. Furthermore, the rapid increase in the use of parallel computing techniques in recent decades has resulted in the publication of a series of relevant scientific papers that exploit this techniques [37,38].

In this paper, the use of a three-stage technique is proposed for the effective training of RBF networks. In the first stage of the technique, the range of values for the parameters of the RBF network is detected. This detection is implemented using the K-Means algorithm [39] for the weights and the variances of the radial functions. After applying the above

procedure, a range of values for the network parameters is created which directly depends on the values produced by K-means algorithm. During the second stage of the proposed work, a global optimization procedure is incorporated to optimize the parameters of the RBF network with respect to equation 3. The training of the parameters is performed inside the interval of values created during the first stage of the technique. In the current work the Genetic Algorithm is used as the method of the second phase, but any optimization technique can be incorporated. Finally, in the third stage of the proposed work, a local optimization procedure is applied to the best solution located in the second phase. The purpose of the present technique is first to identify a reliable range of values for the parameters of RBF networks and then to train the network parameters within this range of values, avoiding possible arithmetic instability problems presented by the established method of training RBF networks.

The remaining of this article is organized as follows: in section 2 the proposed method and the accompanied genetic algorithm are introduced, in section 3 the experimental datasets and the series of experiments conducted are listed and discussed thoroughly followed by section 4 where some conclusions are discussed.

2. Materials and Methods

The three distinct phases of the proposed method are analyzed in this section. The discussion initiates with the first phase, where the construction of the ranges for the parameter values is performed using the K-means algorithm. Subsequently, the steps of the used Genetic Algorithm are presented in detail and finally this section concludes with the description of the final phase, where a local optimization method is applied to the best located chromosome of the second phase.

2.1. The first phase of the proposed method

The method of K-means, used widely in machine learning is incorporated in the first phase for the location of the ranges for the parameters of the RBF network. This method is incorporated to locate the centers and the variances of the possible groups of a series of points. Furthermore, a series of extensions of this method have been published during the past years, such as the Genetic K-means algotithm [41], the unsupervised K-means algorithm [42], the Fixed-centered K-means algorithm [43] etc. A detailed review for the K-means method can be be located in the work of Oti et al. [44]. The K-means method is presented in Algorithm 1 and a graphical rerpesentation is provided in Figure 2.

Algorithm 1 The main steps of the K-means algorithm.

1. **Input:** The set of patterns of the objective problem $(\overrightarrow{x_i})$, $i = 1, ..., M$
2. **Input**: the number of centers k.
3. **Output**: The vectors $\overrightarrow{c_i}$, $i = 1, .., k$ and the quantities σ_i, $i = 1, \ldots, k$
4. **Set** $S_j = \{\}$, $j = 1..k$, as the sets of samples belonging to the same group.
5. **For** each pattern x_i, $i = 1, ..., M$ **do**
   (a) **Set** $j^* = \min_{i=1}^{k} \{D(x_i, c_j)\}$.
   (b) **Set** $S_{j^*} = S_{j^*} \cup \{x_i\}$.
6. **EndFor**
7. **For** each center c_j, $j = 1..k$ **do**
   (a) **Calculate** as M_j the number of points belonging to the group S_j
   (b) **Compute** c_j as

$$c_j = \frac{1}{M_j} \sum_{i=1}^{M_j} x_i$$

8. **EndFor**
9. **Calculate** the quantities s_j as

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

10. **Stop** the algorithm, if there is no change in centers c_j.
11. **Go to** step 5.

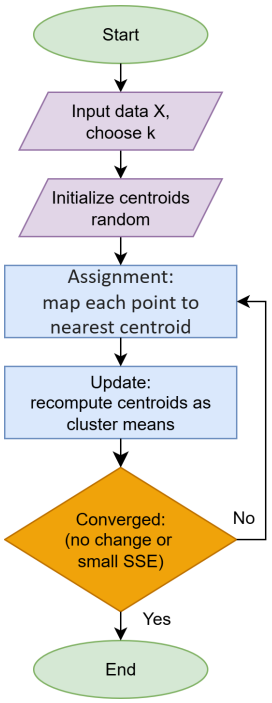

Figure 2. A graphical presentation of the K-means algorithm.

After the calculation of $\overrightarrow{c_i}$, $i = 1, .., k$ and the quantities σ_i, $i = 1, \ldots, k$, the method locates the bound vectors $\overrightarrow{L}$, $\overrightarrow{R}$ for the parameters of the RBF network. The dimension of the bound vectors is defined as:

$$n = (d + 2) \times k \tag{4}$$

For the calculation of the bound vectors the procedure presented in Algorithm 2 is utilized.

Algorithm 2 Algorithm used to obtain the bound vectors $\overrightarrow{L}$, $\overrightarrow{R}$

1. **Input:** The vectors $\overrightarrow{c_i}$, $i = 1, .., k$ and the quantities σ_i, $i = 1, \ldots, k$ of the K-means procedure.
2. **Input**: the initial bound for the weight vector $\overrightarrow{w}$, denoted as $B_w > 0$.
3. **Input**: the scaling factor $F \geq 1$.
4. **Output:** the vectors $\overrightarrow{L}$, $\overrightarrow{R}$.
5. **Set** $m = 0$
6. **For** $i = 1..k$ **do**
   (a) **For** $j = 1..d$ **do**
      i. **Set** L_m=$-F \times c_{ij}$, R_m=$F \times c_{ij}$
      ii. **Set** $m = m + 1$
   (b) **EndFor**
   (c) **Set** $L_m = -F \times \sigma_i$, $R_m = F \times \sigma_i$
   (d) **Set** $m = m + 1$
7. **EndFor**
8. **For** $j = 1, ..., k$ **do**
   (a) **Set** $L_m = -B_w$, $R_m = B_w$
   (b) **Set** $m = m + 1$
9. **EndFor**

2.2. The second phase of the proposed method

During the second phase an optimization procedure is utilized to minimize the equation 3 inside the bound vectors $\overrightarrow{L}$, $\overrightarrow{R}$ of the first phase. In the proposed implementation the Genetic Algorithm was incorporated during the second phase. Genetic algorithm are evolutionary methods, that are based on randomly produced solutions of the objective problem. These solutions are called chromosomes and they are evolved through some operations similar to natural processes of selection, crossover and mutation. Genetic algorithms have been used in a wide series of problems, such as placement of wind turbines [45], water distribution [46], problems appeared in banking transactions [47], optimization of neural networks [48] etc. Also, another advantage of Genetic Algorithms is that they can be easily adopt parallel programming techniques in order to speed up the evolutionary process [49,50]. The layout of the chromosomes used in the obtained genetic algorithm is presented in Figure 3. **I**n this layout the following assumptions are hold:

1. The value $c_{i,j}$ denotes the j element of the i center of the RBF network, with $i \in [1, k]$ and $j \in [1, d]$.
2. The value σ_i represents the σ parameter for the corresponding radial function.
3. The value w_i, $i \in [1, k]$ represents the weight for the corresponding radial function.

Figure 3. The layout of chromosomes used in the second stage of the proposed method.

The steps of the genetic algorithm used in the second phase of the proposed method have as follows:

1. **Initialization step**.
   (a) **Set** the of chromosomes N_c and the maximum number of generations denoted as as N_g.
   (b) **Set** p_s the selection rate and as p_m the mutation rate.

(c) **Initialize** every chromosome $g_i,\ i = 1, \ldots, N_c$ of the population as vector of double numbers. The layout of each chromosome follows the scheme of Figure 3 and the initialization is performed inside the bound vectors $\overrightarrow{L}$, $\overrightarrow{R}$.

(d) **Set** $k = 0$, the generation number.

2. **Fitness calculation step**.

(a) **For** $i = 1, \ldots, N_c$ **do**

i. **Produce** an RBF network $R_i = R(\overrightarrow{x}, \overrightarrow{g_i})$ for the corresponding chromosome $\overrightarrow{g_i}$.

ii. **Estimate** the related fitness value f_i as

$$f_i = \sum_{j=1}^{M} \left(R(\overrightarrow{x}_j, \overrightarrow{g_i}) - y_j\right)^2 \tag{5}$$

(b) **End For**

3. **Genetic operations step**.

(a) Selection procedure: During this procedure the chromosomes are sorted according to their fitness values and the best $p_s \times N_c$ of them are transferred without any change to the next generation. The remaining chromosomes will be substituted by offrpsings produced during crossover and mutation.

(b) Crossover procedure: During this procedure In this procedure $(1 - p_s) \times N_c$ new chromosomes will be created. For each pair $(\tilde{z}, \tilde{w})$ of new chromosomes, two chromosomes (z, w) are selected from the current population using the procedure of tournament selection. The new offsprings are produced following the scheme:

$$\begin{aligned} \tilde{z}_i &= a_i z_i + (1 - a_i) w_i \\ \tilde{w}_i &= a_i w_i + (1 - a_i) z_i \end{aligned} \tag{6}$$

Where the numbers a_i are random numbers having the property $a_i \in [-0.5, 1.5]$ [51].

(c) Mutation procedure: For every element $t_j, j = 1, \ldots, n$ of each chromosome g_i a random number $r \in [0, 1]$ is drawn. If $r \leq p_m$, then this element is altered according to the following scheme:

$$t'_j = \begin{cases} t_j + \Delta(k, R_j - t_j), & t = 0 \\ t_j - \Delta(k, t_j - L_j), & t = 1 \end{cases} \tag{7}$$

The value t is a random number that can be either 0 or 1. The function function $\Delta(k, y)$ is provided by the following equation:

$$\Delta(k, y) = y\left(1 - r^{\left(1 - \frac{k}{N_g}\right)}\right) \tag{8}$$

4. **Termination check step**.

(a) **Set** $k = k + 1$

(b) **If** $k < N_g$ then go to Fitness Calculation Step

(c) **Else** report as outcome of this procedure the best located chromosome g^* with the lowest fitness value.

2.3. The final phase of the proposed method

In the final phase of the proposed method a local optimization procedure is applied to the outcome of the previous phase, in order to locate an actual minimum for the training error of the RBF network. In the current work the BFGS variant of Powell [52] was selected as the local search procedure. This variant can preserve the bounds of the objective function in an efficient way. During the past years a series of modifications for the BFGS method was introduced, such as the limited memory variant L-BFGS ideal for large scale problems [53] or the Regularized Stochastic BFGS Algorithm [54]. Also, Dai published an article on the convergence properties of the BFGS method [55]. The main steps of the final phase of the algorithm have as follows:

1. **Obtain** the best chromosome $\overrightarrow{g^*}$ of the previous phase.
2. **Create** the corresponding RBF network $R^* = R\left(\overrightarrow{x}, \overrightarrow{g^*}\right)$.
3. **Minimize** the training error of the network R^* using the local search procedure of this phase.
4. **Apply** the final network to the test set of the objective problem and report the corresponding test error.

A summary flow chart showing the sequence of the various phases of the proposed work is presented in Figure 4.

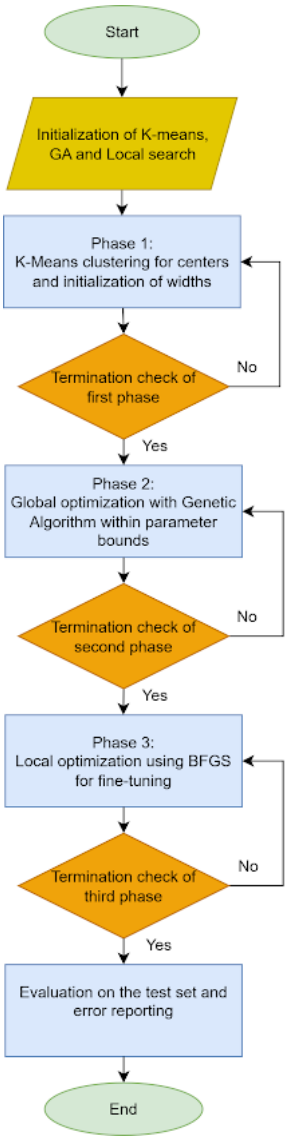

Figure 4. Summary flowchart of the proposed method.

3. Results

The proposed method was validated using a series of classification and regression datasets, obtained from the following online databases:

1. The UCI database, https://archive.ics.uci.edu/(accessed on 29 August 2025)[56]
2. The Keel website, https://sci2s.ugr.es/keel/datasets.php(accessed on 29 August 2025)[57].
3. The Statlib database https://lib.stat.cmu.edu/datasets/index(accessed on 29 August 2025).

3.1. Experimental datasets

The following datasets were incorporated in the experiments conducted in this work:

1. **Appendictis** that is a medical dataset [58].
2. **Alcohol**, which is dataset regarding alcohol consumption [59].
3. **Australian**, which is a dataset produced from various bank transactions [60].
4. **Balance** dataset [61], produced from various psychological experiments.
5. **Cleveland**, a medical dataset which was discussed in a series of papers [62,63].
6. **Circular** dataset, which is an artificial dataset.

7. **Dermatology**, a medical dataset for dermatology problems [64].
8. **Ecoli**, which is related to protein problems [65].
9. **Hayes-roth** dataset [66].
10. **Heart**, which is a dataset related to heart diseases [67].
11. **HeartAttack**, which is a medical dataset for the detection of heart diseases
12. **Housevotes**, a dataset which is related to the Congressional voting in USA [68].
13. **Ionosphere**, a dataset that contains measurements from the ionosphere [69,70].
14. **Liverdisorder**, a medical dataset that was studied thoroughly in a series of papers[71, 72].
15. **Lymography** [73].
16. **Mammographic**, which is a medical dataset used for the prediction of breast cancer [74].
17. **Parkinsons**, which is a medical dataset used for the detection of Parkinson's disease [75,76].
18. **Pima**, which is a medical dataset for the detection of diabetes[77].
19. **Popfailures**, a dataset related to experiments regarding climate [78].
20. **Regions2**, a medical dataset applied to liver problems [79].
21. **Saheart**, which is a medical dataset concerning heart diseases[80].
22. **Segment** dataset [81].
23. **Statheart**, a medical dataset related to heart diseases.
24. **Spiral**, an artificial dataset with two classes.
25. **Student**, which is a dataset regarding experiments in schools [82].
26. **Transfusion**, which is a medical dataset [83].
27. **Wdbc**, which is a medical dataset regarding breast cancer [84,85].
28. **Wine**, a dataset regarding measurements about the quality of wines [86,87].
29. **EEG**, which is dataset regarding EEG recordings [88,89]. From this dataset the following cases were used: Z_F_S, ZO_NF_S, ZONF_S and Z_O_N_F_S.
30. **Zoo**, which is a dataset regarding animal classification [90] .

Moreover a series of regression datasets was adopted in the conducted experiments. The list with the regression datasets has as follows:

1. **Abalone**, which is a dataset about the age of abalones [91].
2. **Airfoil**, a dataset founded in NASA [92].
3. **Auto**, a dataset related to the consumption of fuels from cars.
4. **BK**, which is used to predict the points scored in basketball games.
5. **BL**, a dataset that contains measurements from electricity experiments.
6. **Baseball**, which is a dataset used to predict the income of baseball players.
7. **Concrete**, which is a civil engineering dataset [93].
8. **DEE**, a dataset that is used to predict the price of electricity.
9. **FA** dataset, related to fat measurements.
10. **Friedman**, which is an artificial dataset[94].
11. **FY,** which is a dataset regarding the longevity of fruit flies.
12. **HO**, a dataset located in the STATLIB repository.
13. **Housing**, regarding the price of houses [95].
14. **Laser**, which contains measurements from various physics experiments.
15. **LW**, a dataset regarding the weight of babes.
16. **Mortgage**, a dataset that contains measurements from the economy of USA.
17. **PL** dataset, located in the STALIB repository.
18. **Plastic**, a dataset regarding problems occurred with the pressure on plastics.
19. **Quake**, a dataset regarding the measurements of earthquakes.
20. **SN**, a dataset related to trellising and pruning.

21. **Stock**, which is a dataset regarding stocks.
22. **Treasury**, a dataset that contains measurements from the economy of USA.

3.2. Experimental results

The experiments were performed on a Debian Linux system with 128GB of ram and the code was implemented in ANSI C++ using the OPTIMUS optimization environment [96]. For the validation of the experiments the ten - fold cross validation technique was utilized. For the classification datasets the average classification error, calculated on the corresponding test set, is reported. This quantity is expressed using the following equation:

$$E_C\left(N\left(\vec{x},\vec{w}\right)\right) = 100 \times \frac{\sum_{i=1}^{N}\left(\text{class}\left(N\left(\vec{x_i},\vec{w}\right)\right) - y_i\right)}{N} \tag{9}$$

Where the set T denotes the related test set and it is defined as $T = (x_i, y_i),\ i = 1,\ldots,N$. Similarly, the average regression is reported for the case of regression datasets and it is expressed using the following equation:

$$E_R\left(N\left(\vec{x},\vec{w}\right)\right) = \frac{\sum_{i=1}^{N}\left(N\left(\vec{x_i},\vec{w}\right) - y_i\right)^2}{N} \tag{10}$$

The values for the parameters of the proposed method are listed in Table 1.

Table 1. The values for each parameter of the proposed method.

NAME	MEANING	VALUE
k	Number of radial functions	10
F	Scaling factor	2.0
B_w	Bound value for the weights	10.0
N_c	Number of chromosomes	500
N_g	Maximum number of generations	200
p_s	Selection rate	0.1
p_m	Mutation rate	0.05

In the experimental tables the following notation is used:

1. The column DATASET represents the name of the objective problem.
2. The column BFGS represents the incorporation of the BFGS optimization method [97] for the training of an artificial neural network [98,99] with 10 processing nodes.
3. The column ADAM represents the application of the Adam optimizer [100,101] to train a neural network with 10 hidden nodes.
4. The column RBF-KMEANS stands for the original training of an RBF network.
5. The column NEAT (NeuroEvolution of Augmenting Topologies) [102] stands for the method NEAT incorporated in the training of neural networks.
6. The column GENRBF stands method introduced in [103] for RBF training.
7. The column PROPOSED denotes the incorporation of the current work.
8. The row average denotes the average classification or regression error for all datasets.

Table 2 reports thirty-two classification datasets and six learning methods: BFGS, ADAM, NEAT, RBF-KMEANS, GENRBF, and the new proposed method. The entries are classification error rates; lower values indicate better performance. The last row gives the mean error for each method across all datasets. Based on these means, the proposed method attains the lowest overall error, about 19.45%. All other methods exhibit substantially higher errors: GENRBF 34.89%, ADAM 33.73%, BFGS 33.50%, NEAT 32.77%, and RBF-KMEANS 28.54%. Thus, on average, the proposed method nearly halves the error relative to the alternatives. At the dataset level, the proposed approach often delivers striking gains. For example, on

Spiral it achieves only 13.26% error, whereas all other methods lie around 45–50%. On Australian it yields 22.67% compared with 32–42% for the others. Similarly, on the challenging medical dataset Cleveland, it obtains 50.86% while the others exceed 67%. Comparable advantages are observed on Heart, HeartAttack, Statheart, Wine, and Z_F_S. There are, however, datasets where the proposed method is not the best. On Hayes Roth and Zoo it performs worse than GENRBF or RBF-KMEANS, and on Transfusion the results are broadly similar to the other methods, with no clear advantage. This indicates that although the method does not dominate in every single problem, the overall trend clearly favors it. In summary, the statistical picture suggests that the proposed method is, on average, the most reliable and effective strategy, producing markedly lower classification errors. While it lags behind certain alternatives on a few datasets, its low mean error underscores a more general and robust solution for classification tasks.

Table 2. Experimental results on the classification datasets using the series of machine learning methods used in this article. The number in cells denote average classification error as measured on the corresponding test set.

DATASET	BFGS	ADAM	NEAT	RBF-KMEANS	GENRBF	PROPOSED
Alcohol	41.50%	57.78%	66.80%	49.38%	52.45%	28.57%
Appendicitis	18.00%	16.50%	17.20%	12.23%	16.83%	15.00%
Australian	38.13%	35.65%	31.98%	34.89%	41.79%	22.67%
Balance	8.64%	7.87%	23.14%	33.42%	38.02%	13.11%
Cleveland	77.55%	67.55%	53.44%	67.10%	67.47%	50.86%
Circular	6.08%	19.95%	35.18%	5.98%	21.43%	5.13%
Dermatology	52.92%	26.14%	32.43%	62.34%	61.46%	36.00%
Hayes Roth	37.33%	59.70%	50.15%	64.36%	63.46%	38.31%
Heart	39.44%	38.53%	39.27%	31.20%	28.44%	16.07%
HeartAttack	46.67%	45.55%	32.34%	29.00%	40.48%	19.20%
HouseVotes	7.13%	7.48%	10.89%	6.13%	11.99%	3.65%
Ionosphere	15.29%	16.64%	19.67%	16.22%	19.83%	12.17%
Liverdisorder	42.59%	41.53%	30.67%	30.84%	36.97%	29.29%
Lymography	35.43%	29.26%	33.70%	25.50%	29.33%	24.36%
Mammographic	17.24%	46.25%	22.85%	21.38%	30.41%	17.79%
Parkinsons	27.58%	24.06%	18.56%	17.41%	33.81%	17.53%
Pima	35.59%	34.85%	34.51%	25.78%	27.83%	24.02%
Popfailures	5.24%	5.18%	7.05%	7.04%	7.08%	6.33%
Regions2	36.28%	29.85%	33.23%	38.29%	39.98%	26.29%
Saheart	37.48%	34.04%	34.51%	32.19%	33.90%	28.50%
Segment	68.97%	49.75%	66.72%	59.68%	54.25%	45.00%
Sonar	25.85%	30.33%	34.10%	27.90%	37.13%	22.00%
Spiral	47.99%	48.90%	50.22%	44.87%	50.02%	13.26%
Statheart	39.65%	44.04%	44.36%	31.36%	42.94%	19.67%
Student	7.14%	5.13%	10.20%	5.49%	33.26%	5.23%
Transfusion	25.84%	25.68%	24.87%	26.41%	25.67%	26.04%
Wdbc	29.91%	35.35%	12.88%	7.27%	8.82%	5.54%
Wine	59.71%	29.40%	25.43%	31.41%	31.47%	9.47%
Z_F_S	39.37%	47.81%	38.41%	13.16%	23.37%	3.73%
Z_O_N_F_S	65.67%	78.79%	77.08%	48.70%	68.40%	41.00%
ZO_NF_S	43.04%	47.43%	43.75%	9.02%	22.18%	4.24%
ZONF_S	15.62%	11.99%	5.44%	4.03%	17.41%	1.98%
ZOO	10.70%	14.13%	20.27%	21.93%	33.50%	9.80%
AVERAGE	**33.50%**	**33.73%**	**32.77%**	**28.54%**	**34.89%**	**19.45%**

Table 3 reports twenty-one regression datasets and six machine-learning models: BFGS, ADAM, NEAT, RBF-KMEANS, GENRBF, and the proposed method. The entries

are absolute prediction errors; smaller values indicate better model performance. The last row gives the mean error for each method across all datasets. The analysis of the means clearly shows that the proposed method attains the smallest overall error, 5.87. The remaining methods have substantially higher errors: BFGS 28.82, ADAM 21.39, NEAT 13.99, RBF-KMEANS 9.56, and GENRBF 13.38. This implies that the proposed approach greatly improves accuracy, reducing the average error to nearly half of the best competing model and far more compared with the classical BFGS and ADAM. At the dataset level there are striking gaps. On Housing the proposed method records an error of 15.36, whereas the other models range from 56 to 97. On Stock the contrast is even stronger, with an error of only 1.44 while others reach well above 300. Similarly, on Plastic the error drops to 2.28, while the alternatives lie between 8 and 26. Comparable advantages are observed on Mortgage, Treasury, Concrete, HO, Quake, and PL. There are, however, a few cases where the proposed method is not the best; for example, on FY and SN it shows slightly higher errors than some competitors, though the differences are small, so no clear advantage emerges there. Overall, the evidence indicates that the proposed method is the most reliable and effective option on average, substantially reducing regression error relative to all other techniques. Although it does not lead on every dataset, the large reduction in mean error and the strong results on several difficult problems demonstrate a robust and broadly applicable solution for regression tasks.

Table 3. Experimental results on the regression datasets using a series of machine learning methods. Numbers in cells represent average regression error as measured on the corresponding test set.

DATASET	BFGS	ADAM	NEAT	RBF-KMEANS	GENRBF	PROPOSED
Abalone	5.69	4.30	9.88	7.37	9.98	6.12
Airfoil	0.003	0.005	0.067	0.27	0.121	0.004
Auto	60.97	70.84	56.06	17.87	16.78	8.81
Baseball	119.63	77.90	100.39	93.02	98.91	88.05
BK	0.28	0.03	0.15	0.02	0.023	0.022
BL	2.55	0.28	0.05	0.013	0.005	0.0004
Concrete	0.066	0.078	0.081	0.011	0.015	0.005
Dee	2.36	0.630	1.512	0.17	0.25	0.15
Housing	97.38	80.20	56.49	57.68	95.69	15.36
Friedman	1.26	22.90	19.35	7.23	16.24	5.99
FA	0.426	0.11	0.19	0.015	0.15	0.013
FY	0.22	0.038	0.08	0.041	0.041	0.054
HO	0.62	0.035	0.169	0.03	0.076	0.009
Laser	0.015	0.03	0.084	0.03	0.075	0.016
Mortgage	8.23	9.24	14.11	1.45	1.92	0.23
PL	0.29	0.117	0.098	2.12	0.155	0.023
Plastic	20.32	11.71	20.77	8.62	25.91	2.28
PY	0.578	0.09	0.075	0.012	0.029	0.021
Quake	0.42	0.06	0.298	0.07	0.79	0.036
SN	0.40	0.026	0.174	0.027	0.027	0.026
Stock	302.43	180.89	12.23	12.23	25.18	1.44
Treasury	9.91	11.16	15.52	2.02	1.89	0.47
AVERAGE	**28.82**	**21.39**	**13.99**	**9.56**	**13.38**	**5.87**

Based on the executions carried out with scripts in the R language concerning the processing of the experimental results on the classification datasets, the significance levels of the p parameter were calculated in order to assess the statistical strength of the comparisons between the proposed model and the alternative methods. In Figure 5 these significance levels are illustrated according to the established scale, where ns indicates no statistically significant difference ($p > 0.005$), while the presence of one to four asterisks denotes

increasing statistical strength, from significant $(p < 0.05)$ to very extremely significant $(p < 0.0001)$. The results show that in all comparisons of the proposed model with the other methods, namely BFGS, ADAM, NEAT, RBF-KMEANS, and GENRBF, the significance levels were found at the highest level with $p < 0.0001$. This indicates that the performance differences are not random but instead reflect a very strong superiority of the proposed model over all the other techniques examined.

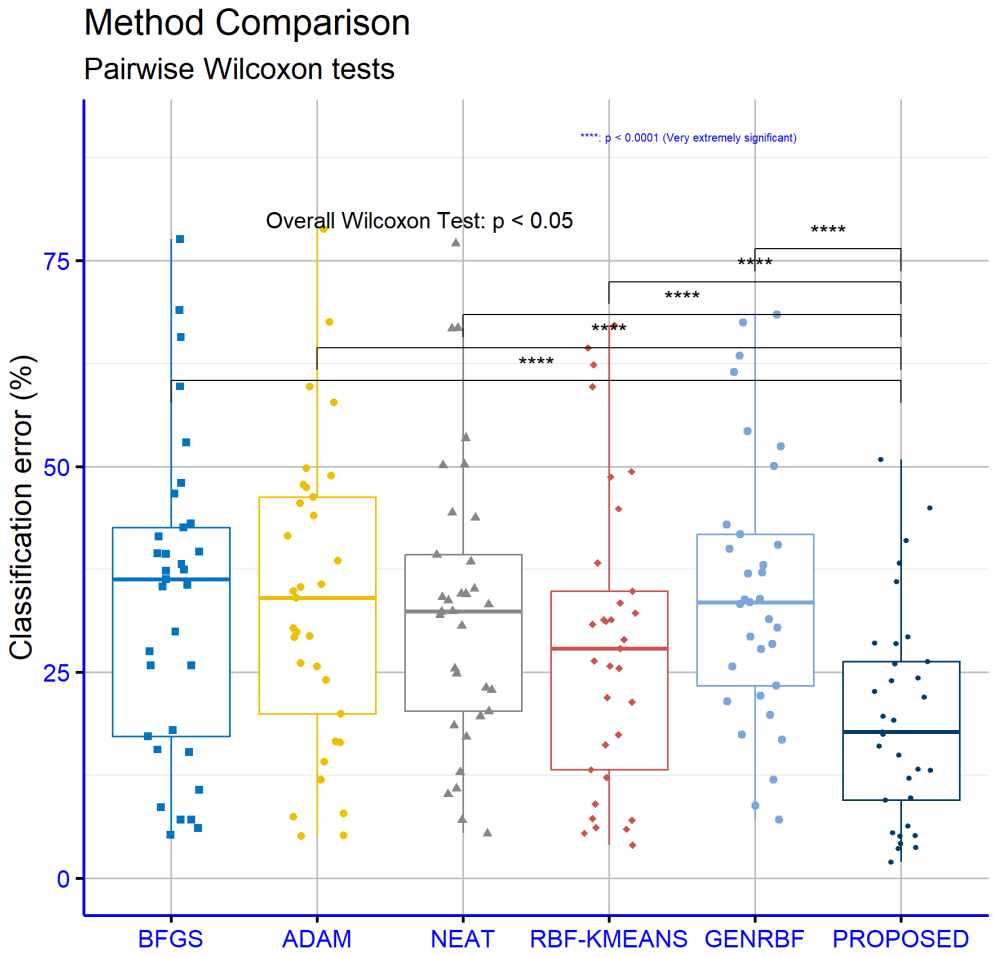

Figure 5. Statistical comparison of results obtained in the classification datasets using a series of machine learning methods.

Similarly, for the regression datasets, the significance levels of the p parameter were computed to assess the statistical strength of the comparisons between the proposed model and the other methods. In Figure 6, the significance levels are illustrated according to the established scale, where two asterisks correspond to high statistical significance $(p < 0.01)$ and four asterisks indicate a very extremely significant difference $(p < 0.0001)$. The results show that, in comparison with the BFGS and ADAM models, the proposed method achieves high statistical significance with $p < 0.01$, demonstrating that its superiority is not due to chance. Furthermore, in all other comparisons, namely against NEAT, RBF-KMEANS, and GENRBF, the significance levels reach the highest level with $p < 0.0001$, which highlights a very strong superiority of the proposed model over the alternative regression methods examined.

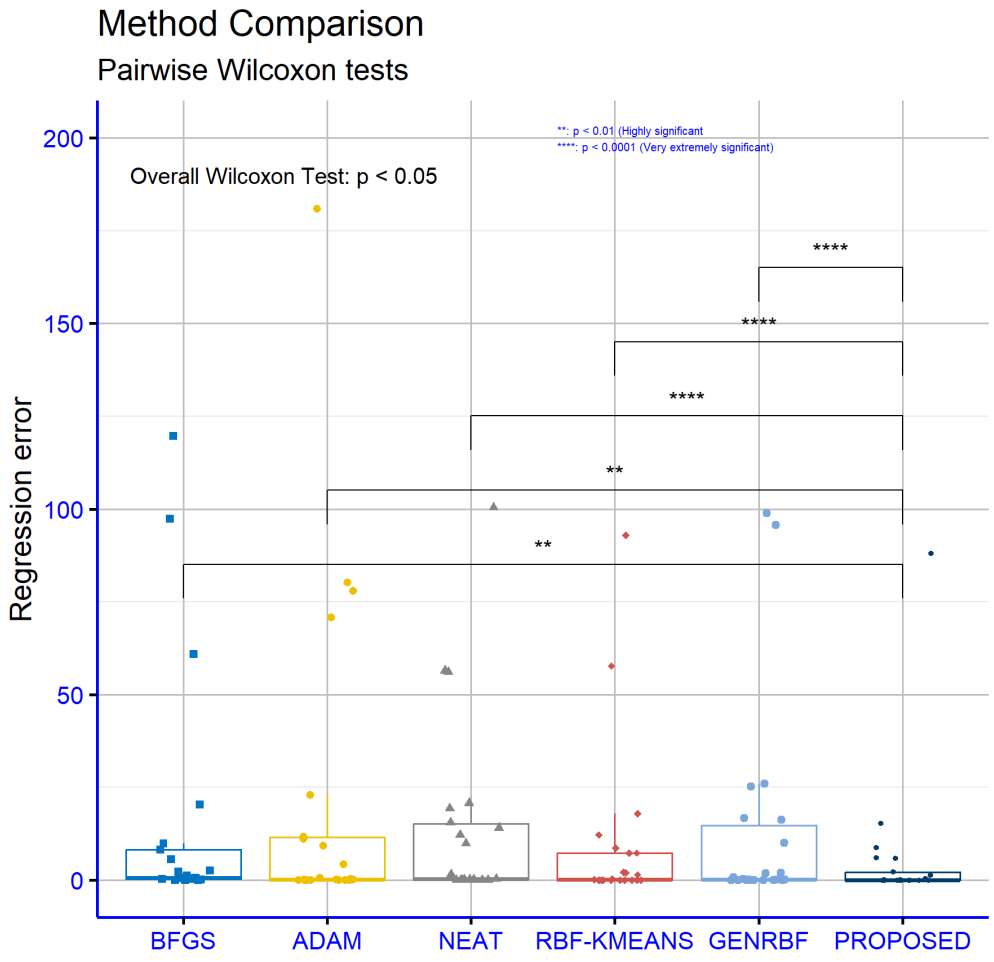

Figure 6. Statistical comparison of the obtained results in the regression datasets using a variety of machine learning methods.

3.3. Experiments with different values of scale factor F

In order to determine the stability of the proposed technique when its critical parameters change, a series of additional experiments were conducted. In one of them, the stability of the technique was studied with the change of the scale factor F. This factor controls the width of the value interval for the network parameters and is a multiple of the initial values estimated by the K-means method of the first phase. In this series of experiments the value of F was altered in the range $[1, 8]$.

Table 4 presents the effect of the scale factor F on the performance of the proposed machine learning model. The parameter F takes four different values, 1, 2, 4, and 8, and for each dataset the classification error rate is reported. Analyzing the mean values, it is observed that $F = 2$ and $F = 4$ achieve the best overall performance, with average errors of 19.45% and 18.53% respectively, compared to 20.99% for $F = 1$ and 18.60% for $F = 8$. This indicates that selecting an intermediate value of the initialization factor improves performance, reducing the error by about two percentage points relative to the baseline case of $F = 1$. At the individual dataset level, interesting patterns emerge. For example, in Sonar the error drops significantly from 32.90% at $F = 1$ to 18.75% at $F = 4$, suggesting that the parameter F strongly influences performance in certain problems. In contrast, in Spiral increasing F worsens the results, as the error rises from 12.03% at $F = 1$ to 23.56% at $F = 8$. Similarly, in the Australian dataset a gradual increase of F from 1 to 8 systematically improves performance, reducing the error from 24.04% to 20.59%. Overall, the data show

that the effect of the scale factor is not uniform across all problems, but the general trend indicates improvement when F increases from 1 to 2 or 4. Choosing $F = 4$ appears to yield the best mean result, although the difference compared with $F = 8$ is very small. Therefore, it can be concluded that tuning this parameter plays an important role in the stability and accuracy of the model, and that intermediate values such as 4 constitute a good general choice.

Table 4. Experimental results on the classification datasets using the proposed method and a series of values for the critical parameter F.

DATASET	$F = 1$	$F = 2$	$F = 4$	$F = 8$
Alcohol	28.83%	28.57%	28.83%	30.09%
Appendicitis	14.60%	15.00%	14.40%	15.50%
Australian	24.04%	22.67%	21.52%	20.59%
Balance	21.03%	13.11%	11.87%	11.44%
Cleveland	50.45%	50.86%	51.59%	50.90%
Circular	4.13%	5.13%	3.67%	3.49%
Dermatology	38.34%	36.00%	35.83%	34.97%
Hayes Roth	51.85%	38.31%	32.62%	33.92%
Heart	17.26%	16.07%	15.63%	15.30%
HeartAttack	22.07%	19.20%	19.30%	19.07%
HouseVotes	4.13%	3.65%	3.39%	4.81%
Ionosphere	14.69%	12.17%	8.83%	7.51%
Liverdisorder	29.35%	29.29%	28.53%	29.23%
Lymography	26.86%	24.36%	18.07%	19.86%
Mammographic	18.21%	17.79%	16.75%	17.05%
Parkinsons	18.32%	17.53%	15.68%	14.05%
Pima	23.53%	24.02%	23.72%	23.26%
Popfailures	7.83%	6.33%	5.15%	4.69%
Regions2	26.27%	26.29%	26.15%	25.73%
Saheart	29.24%	28.50%	28.74%	29.41%
Segment	45.08%	45.00%	42.14%	42.10%
Sonar	32.90%	22.00%	18.75%	18.05%
Spiral	12.03%	13.26%	16.66%	23.56%
Statheart	19.30%	19.67%	20.00%	19.44%
Student	6.33%	5.23%	5.10%	5.55%
Transfusion	25.54%	26.04%	25.66%	24.42%
Wdbc	4.86%	5.54%	5.75%	5.29%
Wine	12.18%	9.47%	8.59%	7.65%
Z_F_S	4.37%	3.73%	3.73%	3.37%
Z_O_N_F_S	39.80%	41.00%	40.04%	40.80%
ZO_NF_S	4.26%	4.24%	4.58%	3.78%
ZONF_S	2.52%	1.98%	2.58%	1.96%
ZOO	12.40%	9.80%	7.60%	6.90%
AVERAGE	**20.99%**	**19.45%**	**18.53%**	**18.60%**

Table 5 shows the effect of the scale factor F on the performance of the proposed regression model. Based on the mean errors, the best overall performance occurs at $F = 4$ with an average error of 5.68, while the values for $F = 1$, $F = 2$ and $F = 8$ are 5.94, 5.87, and 5.78, respectively. The differences across the four settings are not large, but they indicate that intermediate values and especially $F = 4$ tend to offer the best accuracy stability trade-off. At the level of individual datasets, substantial variations are observed. For Friedman the reduction is dramatic, with error dropping from 6.74 at $F = 1$ to 1.41 at $F = 8$, highlighting that proper tuning of F can have a strong impact on performance. Laser shows a similarly large improvement, from 0.027 at $F = 1$ to just 0.0024 at $F = 8$.

Mortgage also improves markedly, from 0.67 at $F = 1$ to 0.015 at $F = 8$. By contrast, in some datasets the value of F has little practical effect, such as Quake and HO, where errors remain nearly constant regardless of F. There are also cases like Housing where increasing F degrades performance, with error rising from 14.64 at $F = 1$ to 18.48 at $F = 8$. Overall, the results indicate that the scale factor F has a significant but nonuniform influence on model performance. In some datasets it sharply reduces error, while in others its impact is negligible or even negative. Nevertheless, the aggregate picture based on the mean errors suggests that $F = 4$ and $F = 8$ yield the most reliable results, with $F = 4$ being the preferred choice for a general-purpose setting.

Table 5. Experimental results on the regression datasets using the proposed technique and a series of values for the parameter F.

DATASET	$F = 1$	$F = 2$	$F = 4$	$F = 8$
Abalone	6.70	6.12	5.70	5.56
Airfoil	0.004	0.004	0.004	0.004
Auto	10.04	8.81	9.82	10.92
Baseball	87.01	88.05	85.87	86.76
BK	0.023	0.022	0.024	0.02
BL	0.01	0.0004	0.0002	0.00007
Concrete	0.008	0.005	0.005	0.006
Dee	0.15	0.15	0.16	0.16
Housing	14.64	15.36	17.34	18.48
Friedman	6.74	5.99	2.06	1.41
FA	0.012	0.013	0.012	0.013
FY	0.055	0.054	0.054	0.053
HO	0.009	0.009	0.01	0.009
Laser	0.027	0.016	0.005	0.0024
Mortgage	0.67	0.23	0.035	0.015
PL	0.023	0.023	0.023	0.022
Plastic	2.32	2.28	2.26	2.22
PY	0.019	0.021	0.013	0.011
Quake	0.036	0.036	0.036	0.036
SN	0.024	0.026	0.025	0.024
Stock	1.69	1.44	1.49	1.48
Treasury	0.57	0.47	0.035	0.031
AVERAGE	**5.94**	**5.87**	**5.68**	**5.78**

In Figure 7, the significance levels are presented for the comparisons between different values of the parameter F in the proposed machine learning method based on the classification datasets. The analysis shows that the comparison between $F = 1$ and $F = 2$ results in high statistical significance with $p < 0.01$, indicating that the transition from the initial value to $F = 2$ has a substantial impact on performance. Similarly, the comparison between $F = 2$ and $F = 4$ also shows high statistical significance with $p < 0.01$, suggesting that further increasing the parameter continues to positively affect the results. However, the comparison between $F = 4$ and $F = 8$ is characterized as not statistically significant, since $p > 0.05$, which means that increasing the parameter beyond $F = 4$ does not bring a meaningful difference in performance. Overall, the findings indicate that smaller values of F play a critical role in improving the model, while increases beyond 4 do not lead to further statistically significant improvements.

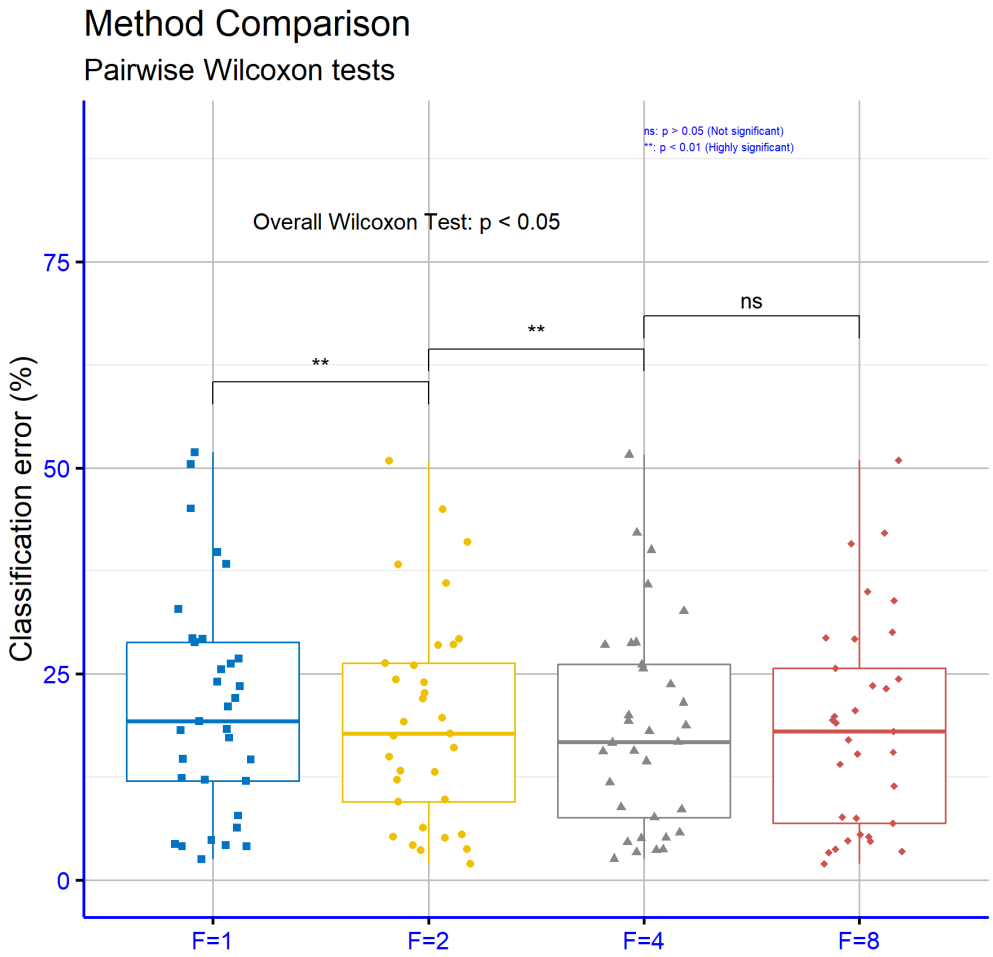

Figure 7. Statistical comparison for the obtained results on the classification datasets using the proposed method and a series of values for the scale parameter F.

In Figure 8, the significance levels are presented for the comparisons between different values of the parameter F in the proposed method based on the regression datasets. The results show that none of the comparisons $F = 1$ vs $F = 2$, $F = 2$ vs $F = 4$, and $F = 4$ vs $F = 8$ exhibit statistically significant differences, since in all cases $p > 0.05$. This means that variations in the parameter F do not substantially affect the performance of the model in regression problems. Therefore, it can be concluded that the choice of the F value is not of critical importance for these datasets and that the model remains stable regardless of the specific setting of this parameter.

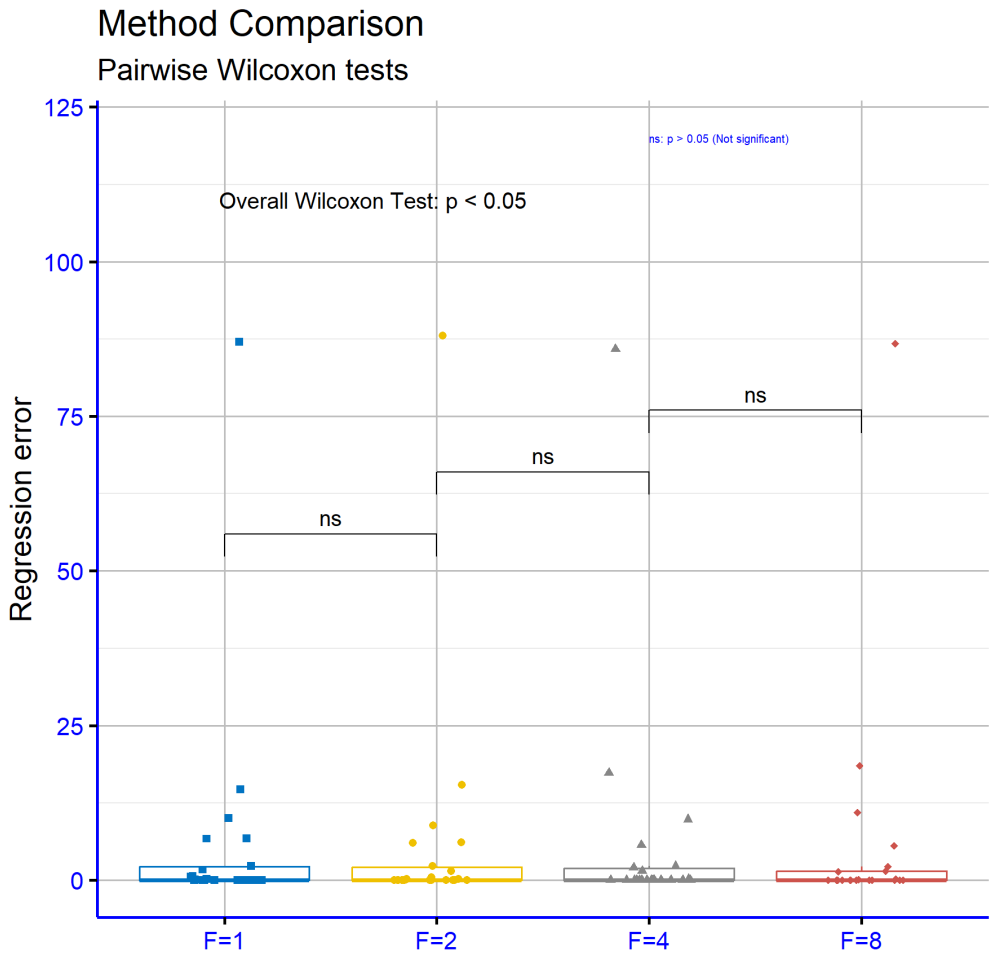

Figure 8. Statistical comparison for the experiments on the regression datasets, using the proposed method and a series of values for the scale parameter F.

3.4. Experiments with differential initialization methods for variances

In another set of experiments, the stability of the proposed method was checked, using a different way of calculating the range of values of the σ parameters of the radial functions. In this work, the value of the variance produced by the K-means algorithm was used as an initial estimate of the σ parameters. This calculation scheme is denoted as σ_1 in the following experimental tables. In this additional set of experiments, two more techniques were used, which will be denoted as σ_{avg} and σ_{max} in the following tables. In the σ_{avg} the following calculation is performed:

$$\sigma_{\text{avg}} = \frac{1}{k}\sum_{i=1}^{k}\sigma_i \tag{11}$$

Subsequently σ_{avg} is used to determine the range of values of the σ parameters of the radial functions of the network. In the σ_{avg} the following quantity is calculated:

$$\sigma_{\text{max}} = \max \sigma_i \tag{12}$$

Afterwards this quantity is used for the determination of the range of values for the σ parameters of the radial functions.

Table 6 presents the effect of three different calculation techniques for the σ parameters used in the radial basis functions of the RBF model. The techniques are a fixed value σ_1, the

mean distance-based initialization $\left(\sigma_{\text{avg}}\right)$, and the maximum distance–based initialization (σ_{max}). Based on the mean errors, the maximum-distance technique yields the lowest overall error at 19.18%. Very close is the mean-distance technique at 19.27%, while the simple σ_1 initialization has a slightly higher error of 19.45%. Although the differences among the three approaches are small, the two adaptive methods $\left(\sigma_{\text{avg}} \text{ and } \sigma_{\text{max}}\right)$ tend to produce marginally better overall performance. At the individual dataset level, behaviors vary. For example, on Wine the σ_{max} choice reduces error to 7.06%, far below the 9.47% obtained with σ_1. On Dermatology, σ_1 performs better than the other two, whereas on Segment the mean-based option is preferable. In some cases the differences are minor e.g., Circular, Pima, and Popfailures where all techniques are comparable; in others the choice of technique materially affects performance, as in Transfusion, where error drops from 26.04% with σ_1 to about 22.78% with the other two methods. Overall, the statistical picture indicates that no single technique dominates across all datasets. Nevertheless, methods that adapt σ to the geometry of the data $\left(\sigma_{\text{avg}} \text{ and } \sigma_{\text{max}}\right)$ tend to yield more reliable and stable results, while the fixed value lags slightly. The average differences are modest, but for certain problems the choice can significantly impact final performance.

Table 6. Experimental results on the classification datasets using the proposed method and a series of techniques for the calculation of the quantities σ used in the radial functions.

DATASET	σ_1	σ_{avg}	σ_{max}
Alcohol	28.57%	28.47%	26.17%
Appendicitis	15.00%	14.20%	15.70%
Australian	22.67%	25.14%	29.96%
Balance	13.11%	12.92%	12.23%
Cleveland	50.86%	51.76%	51.24%
Circular	5.13%	4.78%	4.45%
Dermatology	36.00%	37.54%	37.09%
Hayes Roth	38.31%	38.00%	35.69%
Heart	16.07%	16.52%	15.41%
HeartAttack	19.20%	19.70%	18.97%
HouseVotes	3.65%	3.31%	3.22%
Ionosphere	12.17%	13.00%	12.83%
Liverdisorder	29.29%	28.38%	27.77%
Lymography	24.36%	22.43%	23.50%
Mammographic	17.79%	17.28%	17.41%
Parkinsons	17.53%	14.74%	14.89%
Pima	24.02%	23.28%	23.91%
Popfailures	6.33%	6.37%	6.24%
Regions2	26.29%	25.47%	25.61%
Saheart	28.50%	28.89%	28.28%
Segment	45.00%	43.65%	46.36%
Sonar	22.00%	21.90%	21.30%
Spiral	13.26%	13.73%	13.37%
Statheart	19.67%	20.15%	19.00%
Student	5.23%	5.58%	5.23%
Transfusion	26.04%	22.78%	22.79%
Wdbc	5.54%	5.22%	5.21%
Wine	9.47%	7.93%	7.06%
Z_F_S	3.73%	3.70%	3.73%
Z_O_N_F_S	41.00%	40.20%	41.12%
ZO_NF_S	4.24%	4.42%	4.84%
ZONF_S	1.98%	1.92%	2.06%
ZOO	9.80%	12.50%	10.30%
AVERAGE	**19.45%**	**19.27%**	**19.18%**

Table 7 presents the effect of three different calculation techniques for the σ parameters used in the radial basis functions of RBF model. Based on the mean errors, the distance–average method yields the lowest overall error at 5.81. Very close is the fixed value σ_1 with a mean error of 5.87, while the maximum-distance method shows a slightly higher mean error of 5.96. The difference among the three methods is small, indicating that all can deliver comparable performance at a general level, with a slight advantage for the distance–average approach. At the level of individual datasets, however, significant variations are observed. For example, in Mortgage the σ_{max} method reduces the error dramatically from 0.23 with σ_1 to 0.021, while σ_{avg} also provides a much better result with 0.041. In Treasury the improvement is again substantial, as the error decreases from 0.47 with σ_1 to just 0.08 using σ_{max}. In Stock the reduction is clear, from 1.44 to 1.23, while in Plastic both σ_{avg} and σ_{max} yield lower errors than σ_1. On the other hand, in datasets such as Housing, the use of σ_{max} worsens performance, increasing the error from 15.36 with σ_1 to 19.45. Similarly, in Auto and Baseball the lowest errors are obtained with σ_1, whereas the alternative techniques result in slightly worse performance. Overall, the results show that the choice of calculation technique for σ can significantly affect performance in certain

problems, while in others the difference is negligible. Although no method consistently outperforms the others across all datasets, the distance–average method appears slightly more reliable overall, while the maximum-distance method can in some cases produce very large improvements but in others lead to a degradation of performance.

Table 7. Experimental results on the regression datasets using the proposed method and a series of techniques for the calculation of the quantities σ used in the radial functions.

DATASET	σ_1	σ_{avg}	σ_{max}
Abalone	6.12	6.06	5.43
Airfoil	0.004	0.003	0.003
Auto	8.81	9.80	10.44
Baseball	88.05	86.13	85.89
BK	0.022	0.022	0.022
BL	0.0004	0.008	0.0004
Concrete	0.005	0.005	0.005
Dee	0.15	0.16	0.16
Housing	15.36	15.57	19.45
Friedman	5.99	6.21	6.02
FA	0.013	0.012	0.012
FY	0.054	0.055	0.055
HO	0.009	0.009	0.01
Laser	0.016	0.018	0.011
Mortgage	0.23	0.041	0.021
PL	0.023	0.022	0.022
Plastic	2.28	2.21	2.19
PY	0.021	0.02	0.022
Quake	0.036	0.036	0.036
SN	0.026	0.026	0.025
Stock	1.44	1.32	1.23
Treasury	0.47	0.15	0.08
AVERAGE	**5.87**	**5.81**	**5.96**

In Figure 9, the significance levels are presented for the comparisons of different computation techniques for the σ parameters in the radial basis functions of the proposed machine learning model, based on the classification datasets. The comparisons performed σ_1 vs σ_{avg}, σ_1 vs σ_{max}, and σ_{avg} vs σ_{max} did not show any statistically significant differences, since in all cases $p > 0.05$. This indicates that the choice of compuation method for the σ parameters does not substantially affect the performance of the model on classification problems. Therefore, it can be concluded that the model maintains stable performance regardless of which of the three compuatation techniques is used.

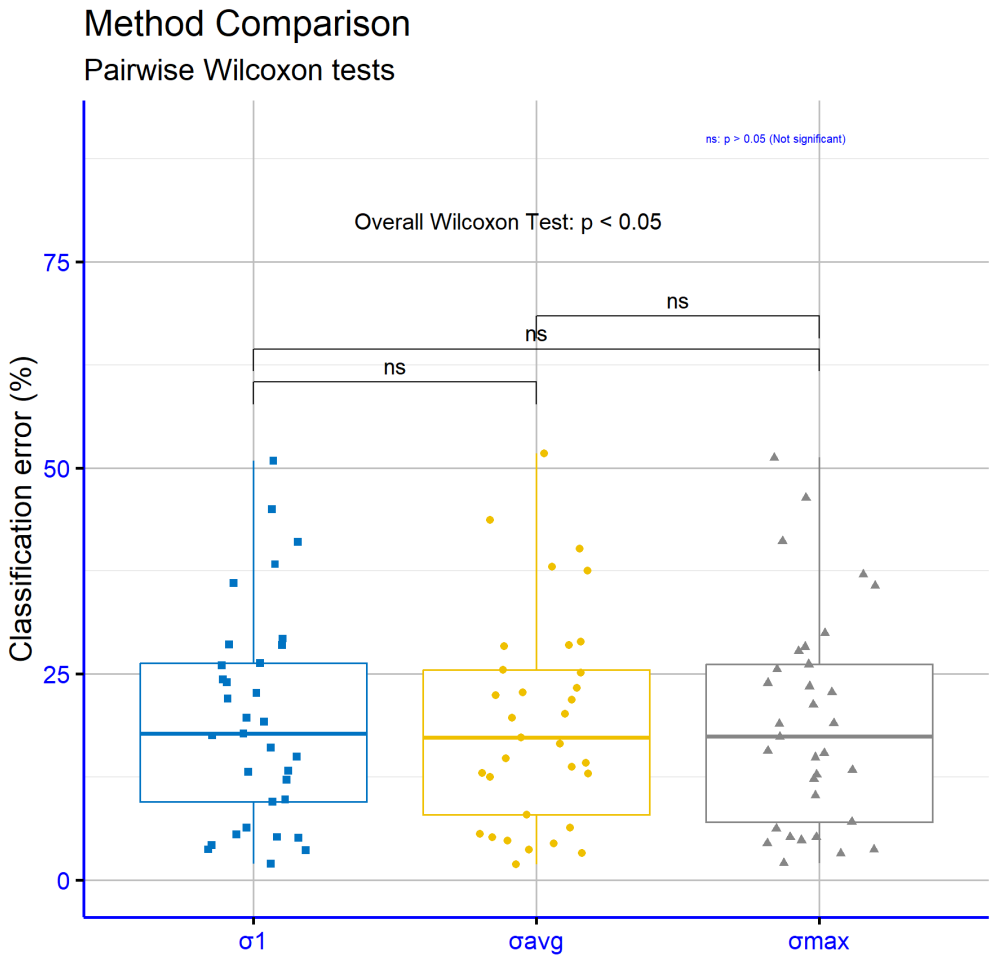

Figure 9. Statistical comparison of the obtained results on the classification datasets, using the proposed method and a series of computation techniques for the range of σ values in the radial functions.

In Figure 10, the significance levels are presented for the comparisons of different computation techniques for the σ parameters in the radial basis functions of the proposed machine learning model, based on the regression datasets. The comparisons examined σ_1 vs σ_{avg}, σ_1 vs σ_{max}, and σ_{avg} vs σ_{max} did not show any statistically significant differences, since in all cases $p > 0.05$. This means that the choice of computation method for the σ parameters does not have a substantial impact on the performance of the model in regression problems. Therefore, it can be concluded that the model demonstrates stable and consistent behavior regardless of which initialization technique is applied.

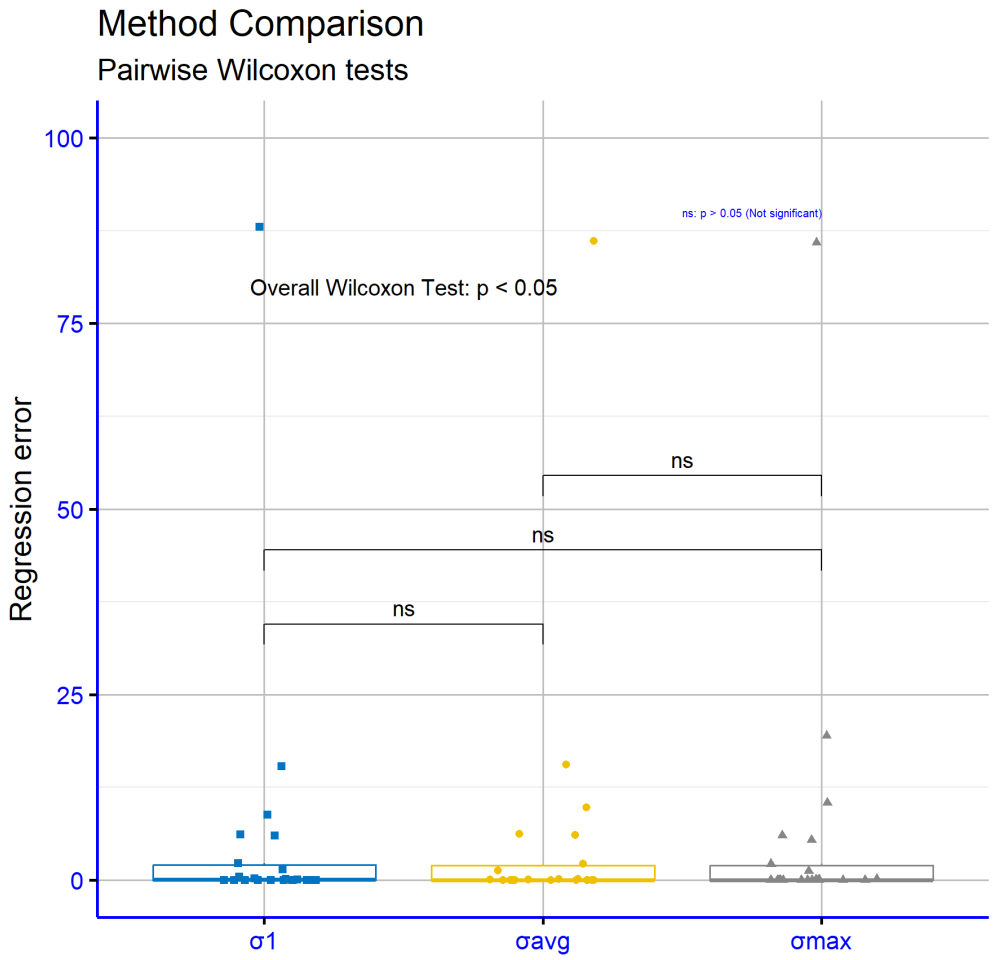

Figure 10. Statistical comparison of the obtained results on the regression datasets, using the proposed method and a series of computation techniques for the range of σ values in the radial functions.

4. Conclusions

The final experimental evidence shows that the three-phase RBF training pipeline bound construction via K-means, global search with a GA inside those bounds, and local refinement with BFGS yields robust gains across heterogeneous classification and regression tasks. On classification, it achieves the lowest mean error (19.45%) with extremely significant superiority over all baselines ($p < 0.0001$), on regression, it attains the smallest mean absolute error (5.87), with $p < 0.01$ against BFGS/ADAM and $p < 0.0001$ against NEAT/RBF-KMEANS/GENRBF. These results indicate that coupling broad exploration with constrained, precise local tuning mitigates numerical instability and local minima, providing reproducible performance improvements.

Sensitivity analyses reveal that the scale factor F materially affects classification at small-to-intermediate settings ($F = 1 \rightarrow 2$ and $F = 2 \rightarrow 4$ are significant at $p < 0.01$), with no meaningful gain from $F = 4$ to $F = 8$, whereas for regression the F comparisons are not significant, highlighting methodological stability. Alternative σ computation methods $\left(\sigma_1, \sigma_{\text{avg}}, \sigma_{\text{max}}\right)$ differ only marginally on average and show no significant differences in either task, reinforcing the method's resilience to low-level design choices.

Automating architecture and hyperparameter adaptation is a natural next step. Joint optimization of the number of RBF units, F, and bounds via Bayesian optimization or meta-learning could reduce manual tuning and improve generalization. Exploring alternative global optimizers (e.g., DE, PSO, CMA-ES) or hybrid GA and Bayesian strategies may

accelerate convergence and enhance exploration, while in the final stage L-BFGS, bound-aware variants, and stochastic formulations could benefit large-scale, high-dimensional settings. A thorough ablation study to quantify each phase's contribution, along with broader post-hoc statistics, would strengthen the evidence base. From a systems perspective, parallel/distributed GA evaluations and GPU-accelerated RBF computations can materially cut runtime. Finally, extending benchmarks to strong non-RBF baselines and integrating the approach into AutoML pipelines together with analyses of interpretability and predictive uncertainty will provide a more complete picture of the method's limits and potential.

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 received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: 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).

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

References

1. M. Mjahed, The use of clustering techniques for the classification of high energy physics data, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **559**, pp. 199-202, 2006.
2. M Andrews, M Paulini, S Gleyzer, B Poczos, End-to-End Event Classification of High-Energy Physics Data, Journal of Physics: Conference Series **1085**, 2018.
3. Viquar, M., Basak, S., Dasgupta, A., Agrawal, S., & Saha, S. (2019). Machine learning in astronomy: A case study in quasar-star classification. Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 3, 827-836.
4. Luo, S., Leung, A. P., Hui, C. Y., & Li, K. L. (2020). An investigation on the factors affecting machine learning classifications in gamma-ray astronomy. Monthly Notices of the Royal Astronomical Society, 492(4), 5377-5390.
5. P. He, C.J. Xu, Y.Z. Liang, K.T. Fang, Improving the classification accuracy in chemistry via boosting technique, Chemometrics and Intelligent Laboratory Systems 70, pp. 39-46, 2004.
6. J.A. Aguiar, M.L. Gong, T.Tasdizen, Crystallographic prediction from diffraction and chemistry data for higher throughput classification using machine learning, Computational Materials Science **173**, 109409, 2020.
7. S.S. Yadav, S.M. Jadhav, Deep convolutional neural network based medical image classification for disease diagnosis. J Big Data **6**, 113, 2019.
8. L. Qing, W. Linhong , D. Xuehai, A Novel Neural Network-Based Method for Medical Text Classification, Future Internet **11**, 255, 2019.
9. I. Kaastra, M. Boyd, Designing a neural network for forecasting financial and economic time series, Neurocomputing **10**, pp. 215-236, 1996.
10. R. Hafezi, J. Shahrabi, E. Hadavandi, A bat-neural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price, Applied Soft Computing **29**, pp. 196-210, 2015.
11. M. Egmont-Petersen, D. de Ridder, H. Handels, Image processing with neural networks—a review, Pattern Recognition **35**, pp. 2279-2301, 2002.
12. G.Peter Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing **50**, pp. 159-175, 2003.
13. M.J. Er, S. Wu, J. Lu, H.L. Toh, Face recognition with radial basis function (RBF) neural networks, IEEE Transactions on Neural Networks **13**, pp. 697-710, 2002.
14. Nam Mai-Duy, Thanh Tran-Cong, Numerical solution of differential equations using multiquadric radial basis function networks, Neural Networks 14, pp. 185-199, 2001.

15. N. Mai-Duy, Solving high order ordinary differential equations with radial basis function networks. Int. J. Numer. Meth. Engng. **62**, pp. 824-852, 2005.
16. Shen, W., Guo, X., Wu, C., & Wu, D. (2011). Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm. Knowledge-Based Systems, 24(3), 378-385.
17. R. -J. Lian, Adaptive Self-Organizing Fuzzy Sliding-Mode Radial Basis-Function Neural-Network Controller for Robotic Systems, IEEE Transactions on Industrial Electronics **61**, pp. 1493-1503, 2014.
18. M. Vijay, D. Jena, Backstepping terminal sliding mode control of robot manipulator using radial basis functional neural networks. Computers & Electrical Engineering **67**, pp. 690-707, 2018.
19. U. Ravale, N. Marathe, P. Padiya, Feature Selection Based Hybrid Anomaly Intrusion Detection System Using K Means and RBF Kernel Function, Procedia Computer Science **45**, pp. 428-435, 2015.
20. M. Lopez-Martin, A. Sanchez-Esguevillas, J. I. Arribas, B. Carro, Network Intrusion Detection Based on Extended RBF Neural Network With Offline Reinforcement Learning, IEEE Access **9**, pp. 153153-153170, 2021.
21. J. A. Leonard and M. A. Kramer, "Radial basis function networks for classifying process faults," in IEEE Control Systems Magazine, vol. 11, no. 3, pp. 31-38, April 1991, doi: 10.1109/37.75576.
22. Ryad Zemouri and Daniel Racoceanu and Noureddine Zerhouni, Recurrent radial basis function network for time-series prediction, Engineering Applications of Artificial Intelligence 16, pp. 453-463, 2003.
23. G. Sideratos and N. Hatziargyriou, "Using Radial Basis Neural Networks to Estimate Wind Power Production," 2007 IEEE Power Engineering Society General Meeting, Tampa, FL, USA, 2007, pp. 1-7, doi: 10.1109/PES.2007.385812.
24. J. Park and I. W. Sandberg, "Universal Approximation Using Radial-Basis-Function Networks," in Neural Computation, vol. 3, no. 2, pp. 246-257, June 1991, doi: 10.1162/neco.1991.3.2.246.
25. L.I. Kuncheva, Initializing of an RBF network by a genetic algorithm, Neurocomputing **14**, pp. 273-288, 1997.
26. F. Ros, M. Pintore, A. Deman, J.R. Chrétien, Automatical initialization of RBF neural networks, Chemometrics and Intelligent Laboratory Systems **87**, pp. 26-32, 2007.
27. D. Wang, X.J. Zeng, J.A. Keane, A clustering algorithm for radial basis function neural network initialization, Neurocomputing **77**, pp. 144-155, 2012.
28. N. Benoudjit, M. Verleysen, On the Kernel Widths in Radial-Basis Function Networks, Neural Processing Letters **18**, pp. 139–154, 2003.
29. E. Ricci, R. Perfetti, Improved pruning strategy for radial basis function networks with dynamic decay adjustment, Neurocomputing **69**, pp. 1728-1732, 2006.
30. Guang-Bin Huang, P. Saratchandran and N. Sundararajan, A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation, IEEE Transactions on Neural Networks **16**, pp. 57-67, 2005.
31. M. Bortman and M. Aladjem, A Growing and Pruning Method for Radial Basis Function Networks, IEEE Transactions on Neural Networks **20**, pp. 1039-1045, 2009.
32. Harpham, C., Dawson, C. W., & Brown, M. R. (2004). A review of genetic algorithms applied to training radial basis function networks. Neural Computing & Applications, 13, 193-201.
33. Sarimveis, H., Alexandridis, A., Mazarakis, S., & Bafas, G. (2004). A new algorithm for developing dynamic radial basis function neural network models based on genetic algorithms. Computers & chemical engineering, 28(1-2), 209-217.
34. Rani R, H. J., & Victoire T, A. A. (2018). Training radial basis function networks for wind speed prediction using PSO enhanced differential search optimizer. PloS one, 13(5), e0196871.
35. Zhang, W., & Wei, D. (2018). Prediction for network traffic of radial basis function neural network model based on improved particle swarm optimization algorithm. Neural Computing and Applications, 29(4), 1143-1152.
36. Qasem, S. N., Shamsuddin, S. M., & Zain, A. M. (2012). Multi-objective hybrid evolutionary algorithms for radial basis function neural network design. Knowledge-Based Systems, 27, 475-497.
37. R. Yokota, L.A. Barba, M. G. Knepley, PetRBF — A parallel O(N) algorithm for radial basis function interpolation with Gaussians, Computer Methods in Applied Mechanics and Engineering **199**, pp. 1793-1804, 2010.
38. C. Lu, N. Ma, Z. Wang, Fault detection for hydraulic pump based on chaotic parallel RBF network, EURASIP J. Adv. Signal Process. **2011**, 49, 2011.
39. MacQueen, J.: Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, No. 14, pp. 281-297, 1967.
40. Ali, H. H., & Kadhum, L. E. (2017). K-means clustering algorithm applications in data mining and pattern recognition. International Journal of Science and Research (IJSR), 6(8), 1577-1584.
41. K. Krishna, M. Narasimha Murty, Genetic K-means algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **29**, pp. 433-439, 1999.
42. K. P. Sinaga, M. -S. Yang, Unsupervised K-Means Clustering Algorithm, IEEE Access **8**, pp. 80716-80727, 2020.

43. M. Ay, L. Özbakır, S. Kulluk, B. Gülmez , G.Öztürk, S. Özer, FC-Kmeans: Fixed-centered K-means algorithm, Expert Systems with Applications **211**, 118656, 2023.
44. E.U. Oti, M.O. Olusola, F.C. Eze, S.U. Enogwe, Comprehensive review of K-Means clustering algorithms, Criterion **12**, pp. 22-23, 2021.
45. S.A. Grady, M.Y. Hussaini, M.M. Abdullah, Placement of wind turbines using genetic algorithms, Renewable Energy **30**, pp. 259-270, 2005.
46. Prasad, T., Park, N., Multiobjective Genetic Algorithms for Design of Water Distribution Networks, J. Water Resour. Plann. Manage. **130**, pp. 73-82, 2004.
47. Sung-Hwan Min, Jumin Lee, Ingoo Han, Hybrid genetic algorithms and support vector machines for bankruptcy prediction, Expert Systems with Applications **31**, pp. 652-660, 2006.
48. D. Whitley, T. Starkweather, C. Bogart, Genetic algorithms and neural networks: Optimizing connections and connectivity, Parallel Computing **14**, pp. 347-361, 1990.
49. Konfrst, Z. (2004, April). Parallel genetic algorithms: Advances, computing trends, applications and perspectives. In 18th International Parallel and Distributed Processing Symposium, 2004. Proceedings. (p. 162). IEEE.
50. Johar, F. M., Azmin, F. A., Suaidi, M. K., Shibghatullah, A. S., Ahmad, B. H., Salleh, S. N., ... & Shukor, M. M. (2013, November). A review of genetic algorithms and parallel genetic algorithms on graphics processing unit (GPU). In 2013 IEEE International Conference on Control System, Computing and Engineering (pp. 264-269). IEEE.
51. P. Kaelo, M.M. Ali, Integrated crossover rules in real coded genetic algorithms, European Journal of Operational Research **176**, pp. 60-76, 2007.
52. M.J.D Powell, A Tolerant Algorithm for Linearly Constrained Optimization Calculations, Mathematical Programming **45**, pp. 547-566, 1989.
53. Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. Mathematical programming, 45(1), 503-528.
54. A. Mokhtari and A. Ribeiro, "RES: Regularized Stochastic BFGS Algorithm," in IEEE Transactions on Signal Processing, vol. 62, no. 23, pp. 6089-6104, Dec.1, 2014, doi: 10.1109/TSP.2014.2357775.
55. Dai, Y. H. (2002). Convergence properties of the BFGS algoritm. SIAM Journal on Optimization, 13(3), 693-701.
56. M. Kelly, R. Longjohn, K. Nottingham, The UCI Machine Learning Repository, https://archive.ics.uci.edu.
57. J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17, pp. 255-287, 2011.
58. Weiss, Sholom M. and Kulikowski, Casimir A., Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems, Morgan Kaufmann Publishers Inc, 1991.
59. Tzimourta, K.D.; Tsoulos, I.; Bilero, I.T.; Tzallas, A.T.; Tsipouras, M.G.; Giannakeas, N. Direct Assessment of Alcohol Consumption in Mental State Using Brain Computer Interfaces and Grammatical Evolution. Inventions 2018, 3, 51.
60. J.R. Quinlan, Simplifying Decision Trees. International Journal of Man-Machine Studies **27**, pp. 221-234, 1987.
61. T. Shultz, D. Mareschal, W. Schmidt, Modeling Cognitive Development on Balance Scale Phenomena, Machine Learning **16**, pp. 59-88, 1994.
62. Z.H. Zhou,Y. Jiang, NeC4.5: neural ensemble based C4.5," in IEEE Transactions on Knowledge and Data Engineering **16**, pp. 770-773, 2004.
63. R. Setiono , W.K. Leow, FERNN: An Algorithm for Fast Extraction of Rules from Neural Networks, Applied Intelligence **12**, pp. 15-25, 2000.
64. G. Demiroz, H.A. Govenir, N. Ilter, Learning Differential Diagnosis of Eryhemato-Squamous Diseases using Voting Feature Intervals, Artificial Intelligence in Medicine. **13**, pp. 147–165, 1998.
65. P. Horton, K.Nakai, A Probabilistic Classification System for Predicting the Cellular Localization Sites of Proteins, In: Proceedings of International Conference on Intelligent Systems for Molecular Biology **4**, pp. 109-15, 1996.
66. B. Hayes-Roth, B., F. Hayes-Roth. Concept learning and the recognition and classification of exemplars. Journal of Verbal Learning and Verbal Behavior **16**, pp. 321-338, 1977.
67. I. Kononenko, E. Šimec, M. Robnik-Šikonja, Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF, Applied Intelligence **7**, pp. 39–55, 1997
68. R.M. French, N. Chater, Using noise to compute error surfaces in connectionist networks: a novel means of reducing catastrophic forgetting, Neural Comput. **14**, pp. 1755-1769, 2002.
69. J.G. Dy , C.E. Brodley, Feature Selection for Unsupervised Learning, The Journal of Machine Learning Research **5**, pp 845–889, 2004.
70. S. J. Perantonis, V. Virvilis, Input Feature Extraction for Multilayered Perceptrons Using Supervised Principal Component Analysis, Neural Processing Letters **10**, pp 243–252, 1999.

71. J. Garcke, M. Griebel, Classification with sparse grids using simplicial basis functions, Intell. Data Anal. **6**, pp. 483-502, 2002.
72. J. Mcdermott, R.S. Forsyth, Diagnosing a disorder in a classification benchmark, Pattern Recognition Letters **73**, pp. 41-43, 2016.
73. G. Cestnik, I. Konenenko, I. Bratko, Assistant-86: A Knowledge-Elicitation Tool for Sophisticated Users. In: Bratko, I. and Lavrac, N., Eds., Progress in Machine Learning, Sigma Press, Wilmslow, pp. 31-45, 1987.
74. M. Elter, R. Schulz-Wendtland, T. Wittenberg, The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process, Med Phys. **34**, pp. 4164-72, 2007.
75. M.A. Little, P.E. McSharry, S.J Roberts et al, Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection. BioMed Eng OnLine **6**, 23, 2007.
76. M.A. Little, P.E. McSharry, E.J. Hunter, J. Spielman, L.O. Ramig, Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. IEEE Trans Biomed Eng. **56**, pp. 1015-1022, 2009.
77. J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, R.S. Johannes, 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, pp.261-265, 1988.
78. D.D. Lucas, R. Klein, J. Tannahill, D. Ivanova, S. Brandon, D. Domyancic, Y. Zhang, Failure analysis of parameter-induced simulation crashes in climate models, Geoscientific Model Development **6**, pp. 1157-1171, 2013.
79. N. Giannakeas, M.G. Tsipouras, A.T. Tzallas, K. Kyriakidi, Z.E. Tsianou, P. Manousou, A. Hall, E.C. Karvounis, V. Tsianos, E. Tsianos, A clustering based method for collagen proportional area extraction in liver biopsy images (2015) Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2015-November, art. no. 7319047, pp. 3097-3100.
80. T. Hastie, R. Tibshirani, Non-parametric logistic and proportional odds regression, JRSS-C (Applied Statistics) **36**, pp. 260–276, 1987.
81. M. Dash, H. Liu, P. Scheuermann, K. L. Tan, Fast hierarchical clustering and its validation, Data & Knowledge Engineering **44**, pp 109–138, 2003.
82. P. Cortez, A. M. Gonçalves Silva, Using data mining to predict secondary school student performance, In Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) (pp. 5–12). EUROSIS-ETI, 2008.
83. I-Cheng Yeh, King-Jang Yang, Tao-Ming Ting, Knowledge discovery on RFM model using Bernoulli sequence, Expert Systems with Applications **36**, pp. 5866-5871, 2009.
84. Jeyasingh, S., & Veluchamy, M. (2017). Modified bat algorithm for feature selection with the Wisconsin diagnosis breast cancer (WDBC) dataset. Asian Pacific journal of cancer prevention: APJCP, 18(5), 1257.
85. Alshayeji, M. H., Ellethy, H., & Gupta, R. (2022). Computer-aided detection of breast cancer on the Wisconsin dataset: An artificial neural networks approach. Biomedical signal processing and control, 71, 103141.
86. M. Raymer, T.E. Doom, L.A. Kuhn, W.F. Punch, Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society, **33** , pp. 802-813, 2003.
87. P. Zhong, M. Fukushima, Regularized nonsmooth Newton method for multi-class support vector machines, Optimization Methods and Software **22**, pp. 225-236, 2007.
88. R. G. Andrzejak, K. Lehnertz, F.Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state," Physical Review E, vol. 64, no. 6, Article ID 061907, 8 pages, 2001.
89. A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis, "Automatic Seizure Detection Based on Time-Frequency Analysis and Artificial Neural Networks," Computational Intelligence and Neuroscience, vol. 2007, Article ID 80510, 13 pages, 2007. doi:10.1155/2007/80510
90. M. Koivisto, K. Sood, Exact Bayesian Structure Discovery in Bayesian Networks, The Journal of Machine Learning Research **5**, pp. 549–573, 2004.
91. 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 Fisheries Division; Technical Report No. 48; Department of Primary Industry and Fisheries, Tasmania: Hobart, Australia, 1994; ISSN 1034-3288
92. Brooks, T.F.; Pope, D.S.; Marcolini, A.M. Airfoil Self-Noise and Prediction. Technical Report, NASA RP-1218. July 1989. Available online: https://ntrs.nasa.gov/citations/19890016302 (accessed on 14 November 2024).
93. I.Cheng Yeh, Modeling of strength of high performance concrete using artificial neural networks, Cement and Concrete Research. **28**, pp. 1797-1808, 1998.
94. Friedman, J. (1991): Multivariate Adaptative Regression Splines. Annals of Statistics, 19:1, 1--141.
95. D. Harrison and D.L. Rubinfeld, Hedonic prices and the demand for clean ai, J. Environ. Economics & Management **5**, pp. 81-102, 1978.
96. I.G. Tsoulos, V. Charilogis, G. Kyrou, V.N. Stavrou, A. Tzallas, Journal of Open Source Software **10**, 7584, 2025.

97. Yuan, Y. X. (1991). A modified BFGS algorithm for unconstrained optimization. IMA Journal of Numerical Analysis, 11(3), 325-332.
98. C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
99. G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control Signals and Systems **2**, pp. 303-314, 1989.
100. D. P. Kingma, J. L. Ba, ADAM: a method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), pp. 1–15, 2015.
101. Y. Xue, Y. Tong, F. Neri, An ensemble of differential evolution and Adam for training feed-forward neural networks. Information Sciences **608**, pp. 453-471, 2022.
102. K. O. Stanley, R. Miikkulainen, Evolving Neural Networks through Augmenting Topologies, Evolutionary Computation **10**, pp. 99-127, 2002.
103. S. Ding, L. Xu, C. Su et al, An optimizing method of RBF neural network based on genetic algorithm. Neural Comput & Applic 21, pp. 333–336, 2012.