applied sciences

Article

A Two-Phase Evolutionary Method to Train RBF Networks

Ioannis G. Tsoulos *, Alexandros Tzallas and Evangelos Karvounis

Department of Informatics and Telecommunications, University of Ioannina, 451 10 Ioannina, Greece; tzallas@uoi.gr (A.T.); ekarvounis@uoi.gr (E.K.)
* Correspondence: itsoulos@uoi.gr

Abstract: This article proposes a two-phase hybrid method to train RBF neural networks for classification and regression problems. During the first phase, a range for the critical parameters of the RBF network is estimated and in the second phase a genetic algorithm is incorporated to locate the best RBF neural network for the underlying problem. The method is compared against other training methods of RBF neural networks on a wide series of classification and regression problems from the relevant literature and the results are reported.

Keywords: RBF networks; classification; regression; genetic algorithms

check for updates

Citation: Tsoulos, I.G.; Tzallas, A.; Karvounis, E. A Two-Phase Evolutionary Method to Train RBF Networks. *Appl. Sci.* **2022**, *12*, 2439. https://doi.org/10.3390/app12052439

Academic Editors: Stylianos Pappas and Sokratis Katsikas

Received: 19 January 2022
Accepted: 24 February 2022
Published: 25 February 2022

1. Introduction

In machine learning, many practical problems appear such as classification and regression problems. A good programming tool that can be used to tackle this problem is radial basis function (RBF) networks [1]. These networks typically are expressed as a function:

$$y(x) = \sum_{i=1}^{k} w_i \phi(\|x - c_i\|) \tag{1}$$

where $\overrightarrow{x}$ is the input pattern, the vector $\overrightarrow{w}$ is called the weight vector and $y(x)$ is the predicted value of the network. RBF networks are feedforward neural networks [2] with three computational layers:

1. The input layer, where the problem is presented in the form of patterns to the neural network.
2. The processing layer, where a computation is performed using the Gaussian processing units $\phi(x)$. These units can have many forms in the relevant literature but the most used form is the Gaussian function expressed as:

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

   The value $\phi(x)$ depends only on the distance of vector $\overrightarrow{x}$ from some other vector $\overrightarrow{c}$, which typically is called centroid.
3. The output layer where the output of every function $\phi(x)$ is multiplied by a corresponding weight value w_i.

RBF networks have been used in many classification and regression problems from the areas of physics [3–6], medicine [7–9], solution of differential equations [10,11], chemistry [12–14], economics [15–17], digital communications [18,19], etc. Furthermore, recently, RBF networks have been used in more difficult problems such as the authentication assurance of meat products [20], trajectory tracking for electrohydraulic servo systems, identification of geographical origin for foods [21], prediction of solution gas–oil ratio of crude oil systems [22], prediction of occurrences of haloketones in tap water [23], health monitoring [24], etc. Because of the extensive use of RBF networks, many methods have been proposed in the recent literature to enhance them. There are methods that parallelize the RBF networks [25,26], methods that improve the initialization of the RBF parameters [27–29], methods that alter the architecture of the network [30–32], methods aimed to locate the best set of RBF parameters with global optimization techniques [33–35], etc. This article transforms the problem of RBF training into an optimization problem and applies a modified genetic algorithm technique to solve it. The global optimization problem is defined as:

$$\min(E(y)) = \sum_{i=1}^{m} (y(x_i) - t_i)^2 \tag{3}$$

where m is the total number of input patterns and t_i is the output for pattern x_i. The suggested approach has two phases: firstly, reasonable bounds for the RBF parameters are estimated using the k-means [36] algorithm and in the second phase, the modified algorithm is used to solve the problem of Equation (3) inside the bounds located in the first phase.

The rest of this paper is organized as follows: in Section 2 the proposed method is described, in Section 3 the conducted experiments are listed and the proposed method is compared against the traditional training of RBF networks and finally, in Section 4 some conclusions are derived.

2. Method Description

The proposed method can be divided into two main phases: during the first phase, an approximation for the bound of RBF parameters is made using the k-means algorithm; in the second phase, the optimization problem is solved using a modified genetic algorithm. These phases are outlined in detail in the following subsections.

2.1. Bound Location Phase

The proposed genetic algorithm has chromosomes with dimension $(d+1) \times k$, where d is the dimension of the input problem, i.e., the dimension of the vector $\overrightarrow{x_i}$ in Equation (3), and k is the total number of processing units of the RBF network. The layout of each chromosome is presented in Table 1. Every center $\overrightarrow{c_i}$ in Equation (1) is a vector of dimension d and an additional parameter is also reserved for the parameter σ of every $\phi(x)$ function. The centroids and the corresponding variances are estimated using the k-means algorithm that is described in Algorithm 1. The value σ_i for every $\phi_i(x)$ is calculated as:

$$\sigma_i = \sum_{j=1}^{d} s_{ij}^2 \tag{4}$$

After the estimation of c_i and σ_i, the vectors $\overrightarrow{L}$, $\overrightarrow{R}$ with dimension $(d+1) \times k$ are constructed. These vectors will serve as the bounds for the chromosomes of the genetic population. These vectors are constructed using the following procedure:

```
1.  Set $m = 0$
2.  Set $F > 1$
3.  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 $m = m + 1$
4.  EndFor
```

Table 1. The layout of the chromosomes in the proposed genetic algorithm.

c_{11}	c_{12}	...	c_{1d}	σ_1	c_{21}	c_{22}	...	c_{2d}	σ_2	...	c_{k1}	c_{k2}	...	c_{kd}	σ_k

Algorithm 1: The k-means algorithm.

```
1. Repeat
   (a) $S_j = \{\},\ j = 1..k$
   (b) For every sample $x_i$ Do
       i.  Set $j^* = \min_{i=1}^{k}\{D(x_i, c_j)\}$, where $j^*$ is the nearest center for sample $x_i$.
       ii. Set $S_{j^*} = S_{j^*} \cup \{x_i\}$.
   (c) EndFor
   (d) For every center $c_j$ Do
       i.  Set $M_j$ = number of elements in $S_j$
       ii. Update $c_j$
               $c_j = \frac{1}{M_j}\sum_{i=1}^{M_j} x_i$
   (e) EndFor
2. Calculate the corresponding variances
               $s_j^2 = \frac{\sum_{i=1}^{M_j}(x_i - c_j)^2}{M_j}$
3. Terminate when $c_j$ no longer changes.
```

2.2. Main Algorithm

The genetic algorithm used here is based on the algorithm denoted as GA(c_{r1}, l) in the Kaelo and Ali's paper [37], with a modified stopping rule as proposed in [38]. The basic steps of the main algorithm are given below:

1. **Initialization Step**
   (a) **Read** the train set with m patterns of d dimension.
   (b) **Set** k the number of nodes for the RBF network.
   (c) **Estimate** the vectors $\overrightarrow{L}$, $\overrightarrow{R}$ using the procedure of Section 2.1.
   (d) **Initialize** a genetic population of N_C random chromosomes inside $[L, R]$.
   (e) **Set** the selection rate $p_s \in [0,1]$, the mutation rate $p_M \in [0,1]$, $iter = 0$ and i_{max} the maximum number of generations.
2. **Evaluation Step**
   For every chromosome g **calculate** the fitness f_g using the procedure defined in Section 2.3 .
3. **Genetic operations step**
   During this step three genetic operations are performed: selection, crossover and mutation.
   (a) **Selection procedure.** Firstly, the chromosomes are sorted with relevance to their corresponding fitness value. The best $(1 - p_s) \times N_c$ are transferred without change to the next generation and the remaining ones are substituted by offspring created through the crossover procedure. In the crossover procedure, the mating parent are selected using a tournament selection for every parent. The tournament selection is as follows:
      i. Select a set of $T > 2$ chromosomes from the population.
      ii. Return the chromosome with the best fitness value in that subset.

   (b) **Crossover procedure**. For every pair (z, w) of selected parents create two new offspring $\tilde{z}$ and $\tilde{w}$:

$$\begin{aligned} \tilde{z}_i &= a_i z_i + (1 - a_i) w_i \\ \tilde{w}_i &= a_i w_i + (1 - a_i) z_i \end{aligned} \tag{5}$$

   with a_i a random number and $a_i \in [-0.5, 1.5]$ [37]. This crossover scheme will be able to better explore the search space of the train error.

   (c) **Mutation procedure**. For every element of each chromosome, create a random number $r \in [0, 1]$. If $r \leq p_m$, then change that element randomly. The mutation is performed in a way similar to other approaches of genetic algorithms [38] and it is described in Section 2.5.

   (d) **Replace** the $p_s \times N_c$ worst chromosomes in the population with the generated offsprings.

4. **Termination Check Step**
   (a) **Set** $iter = iter + 1$
   (b) **Terminate** if the termination criteria of Section 2.4 are satisfied, **else Goto** Evaluation Step.

2.3. Fitness Evaluation

In this step, a valid RBF network $y(x) = \sum_{i=1}^{k} w_i \phi(\|x - c_i\|)$, is created using the chromosome g and is subsequently trained using the typical training procedure for RBF networks. The main steps to calculate the fitness f_g of a chromosome g are the following:

1. **Decode** the chromosome g to the parts (centers and variances) of the RBF network as defined by the layout of Table 1.
2. **Calculate** the output vectors $w_1, w_w, \ldots, w_k$ by solving the induced system of equations:
   (a) **Set** $W = w_{kj}$ the matrix of k weights, $\Phi = \phi_j(x_i)$ and $T = \{t_i\}$.
   (b) **Solve:**

$$\Phi^T \left(T - \Phi W^T \right) = 0 \tag{6}$$

$$W^T = \left(\Phi^T \Phi \right)^{-1} \Phi^T T = \Phi^\dagger T \tag{7}$$

   The matrix $\Phi^\dagger = \left(\Phi^T \Phi \right)^{-1} \Phi^T$ is the pseudo-inverse of Φ, with

$$\Phi^\dagger \Phi = I \tag{8}$$

3. **Set** $f_g = \sum_{i=}^{m} (y(x_i) - t_i)^2$

2.4. Stopping Rule

Define as g_{best} the best chromosome in the population and define as $\sigma^{(\text{iter})}$ the variance of best fitness $f(g_{\text{best}})$ at generation iter. If fitness $f(g_{\text{best}})$ has not improved for a number of generations, then probably the algorithm should terminate. Hence, the termination rule is defined as:

$$\text{iter} \geq i_{\max} \text{ OR } \sigma^{(\text{iter})} \leq \frac{\sigma^{(\text{klast})}}{2} \tag{9}$$

where klast is the last generation where a new minimum was found.

2.5. Mutation Procedure

Let $w = (w_1, w_2, ..., w_n)$ be the chromosome to be mutated. The proposed mutation procedure modifies w_i with probability p_m and the resulting element w_i' is given by

$$w_i' = \begin{cases} w_i + \Delta(\text{iter}, R_i - w_i), \text{ if} & t > \frac{1}{2}0 \\ w_i - \Delta(\text{iter}, w_i - L_i), \text{ otherwise} & \end{cases} \tag{10}$$

where t is a random number with $t \in [0, 1]$. The function $\Delta(\text{iter}, y)$ is given by:

$$\Delta(\text{iter}, y) = y \left(1 - r^{\left(1 - \frac{\text{iter}}{\text{ITERMAX}}\right)^b} \right) \tag{11}$$

where r is a random number in $[0, 1]$ and b controls the change of element w_i. In the proposed algorithm the value $b = 5$ was used.

3. Experiments

In order to evaluate the performance of the proposed method, comparative experiments were performed on a series of well-known classification and regression datasets from the relevant literature.

3.1. Experimental Setup

The RBF network was coded in ANSI C++, using the Armadillo library [39] and the optimization was performed using the genetic optimization method of the optimization package OPTIMUS, that is freely available from https://github.com/itsoulos/OPTIMUS/ (accessed on 18 January 2021). Furthermore, to have more reliability in the results the commonly used method of 10-fold cross-validation was used, which means that the original data were randomly partitioned into 10 equally sized subsamples. Subsequently, 10 independent experiments were conducted: in each experiment one subsample was used as the testing data and all the others as the training data. The average error on the test data was the total test error. All the experiments were executed 30 times with different initialization for the random generator each time. The random generator used was the function drand48() of the C programming language. The execution environment was an Intel Xeon E5-2630 multicore machine using the OpenMP library [40] for parallelization and the Ubuntu Linux operating system. The parameters for the genetic algorithm are displayed in Table 2. The parameters of the method were chosen so that there is a balance between speed and efficiency of the method.

Table 2. Experimental parameters.

Parameter	Value
k	10
N_c	200
p_s	0.90
p_m	0.05
F	3.0
i_{max}	200

3.2. Experimental Datasets

The classification problems used for the experiments were found in most cases on two Internet databases:

1. UCI dataset repository, https://archive.ics.uci.edu/ml/index.php (accessed on 18 January 2021)

2. Keel repository, https://sci2s.ugr.es/keel/datasets.php (accessed on 18 January 2021) [41].

The following classification datasets were used:

1. The **Alcohol** dataset, which is related to alcohol consumption [42].
2. The **Appendicitis** dataset, proposed in [43].
3. The **Australian** dataset [44], which refers to credit card applications.
4. The **Balance** dataset [45], which is used to predict psychological states.
5. The **Cleveland** dataset, used in various papers to detect heart disease [46,47].
6. The **Dermatology** dataset [48], which is used for differential diagnosis of erythematosquamous diseases.
7. The **Glass** dataset, which contains glass component analysis and has been used in a variety of papers [49,50].
8. The **Hayes roth** dataset [51].
9. The **Heart** dataset [52], used to detect heart disease.
10. The **HouseVotes** dataset [53], which is about votes in the U.S. House of Representatives.
11. The **Ionosphere** dataset, which contains data from the Johns Hopkins Ionosphere database and has been studied in a number of papers [54,55].
12. The **Liverdisorder** dataset [56], used to detect liver disorders in people using blood analysis.
13. The **Mammographic** dataset [57], used to identify the severity of a mammographic mass lesions.
14. The **Parkinson** dataset, which is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD) [58].
15. The **Pima** dataset [59], used to detect the presence of diabetes.
16. The **Popfailures** dataset [60], related to climate model simulation crashes.
17. The **Regions2** dataset, which was created from liver biopsy images of patients with hepatitis C [61].
18. The **Ring** dataset [62], a 20-dimension problem with two classes, where each class is drawn from a multivariate normal distribution.
19. The **Saheart** dataset [63], used to detect heart disease.
20. The **Segment** dataset [64], which contains patterns from a database of seven outdoor images (classes).
21. The **Sonar** dataset [65]. The task is to discriminate between sonar signals bounced off a metal cylinder.
22. The **Spiral** dataset, which is an artificial dataset containing 1000 two-dimensional examples that belong to two classes.
23. The **Tae** dataset [66], which concerns evaluations of teaching performance.
24. The **Thyroid** dataset, which concerns thyroid disease records [67].
25. The **Wdbc** dataset [68], which contains data from breast tumors.
26. The **Wine** dataset, used to detect through chemical analysis the origin of wines in various research papers [69,70].
27. The **EEG** dataset from [71], which consists of five sets (denoted as Z, O, N, F and S), each containing 100 single-channel EEG segments each lasting 23.6 s. With different combinations of these sets the produced datasets are Z_F_S, ZO_NF_S, ZONF_S.
28. The **Zoo** dataset [72], where the task is to classify animals in seven predefined classes.

The regression datasets are in most cases available from the StatLib URL ftp://lib.stat.cmu.edu/datasets/index.html (accessed on 18 January 2021):

1. The **Abalone** dataset [73], which can be used to obtain a model to predict the age of abalone from physical measurements.
2. The **Airfoil** dataset, which is used by NASA for a series of aerodynamic and acoustic tests [74].

3. The **Anacalt** dataset [75], which contains information about the decisions taken by a supreme court.
4. The **BK** dataset [76], used to estimate the points in a basketball game.
5. The **BL** dataset, which can be downloaded from StatLib and contains data from an experiment on the effects of machine adjustments on the time it takes to count bolts.
6. The **Concrete** dataset, used to measure the concrete compressive strength [77].
7. The **Housing** dataset, taken from the StatLib library, which is maintained at Carnegie Mellon University, and described in [78].
8. The **Laser** dataset, used in laser experiments. It has been obtained from the Santa Fe Time Series Competition Data repository.
9. The **MB** dataset, available from Smoothing Methods in Statistics [77].
10. The **NT** dataset, which contains data from [79] that examined whether the true mean body temperature is 98.6 F.
11. The **Quake** dataset, whose objective is to approximate the strength of a earthquake. It has been obtained from the Bilkent University Function Approximation Repository.

3.3. Experimental Results

The results for the classification datasets are listed in Table 3 and for the regression datasets the results are reported in Table 4. For the first case, the average classification error is reported and for the case of regression datasets, the total test error is reported. The column KRBF denotes the classic RBF training method, GRBF denotes the method proposed in [80] and the column "proposed" denotes the proposed method. The KBF simply consists of two phases: in the first phase, the centers and variances are estimated through the k-means algorithm and in the second phase, a system of equations is solved to obtain the weights w_i of the RBF network.

From the experimental results, it is clear that the proposed method is significantly superior to other methods in almost all datasets. In the proposed method, the appropriate initialization interval was found for the parameters of RBF using k-means. A parallel genetic algorithm was then applied to this previous value range, creating a variety of neural networks. This combination of techniques obviously has very good results as it combines a very efficient clustering method and an excellent optimization method that is ideally parallelized. Of course, the new method requires much more execution time, due to the presence of the genetic algorithm, but the parallel execution of the software drastically reduces this time. Furthermore, in order to study the effectiveness of the selection of parameter F an additional experiment was conducted, where the best fitness of the genetic algorithm is plotted for the Wine problem. The outcome of this experiment is graphically outlined in Figure 1. The graph shows that the behavior of the proposed method does not change significantly for different values of the parameter F. Furthermore, the plot for the Wine dataset of best, worst and average fitness for $F = 3$ is shown in Figure 2. An additional experiment was performed to evaluate the effect of the parameter change k on the results. In Figure 3, the plot for different values of k for the Housing dataset is outlined and in Figure 4 the same experiment is shown for the Z_F_S classification dataset. Of course, from the value $k = 4$ onwards, the error falls but not at the same rate. The proposed value $k = 10$ was used in all datasets in order to have a balance between the speed and the efficiency of the method. Finally, the classification performance was evaluated based on two evaluation metrics: precision and recall for some datasets as shown in Table 5. In these results, the precision of the proposed method showed the best classification performance under different datasets. We found the same trends in recall metric in two of three datasets (spiral and EEG datasets).

Table 3. Classification error for different datasets.

Dataset	**KRBF**	**GRBF**	**Proposed**
Alcohol	46.63%	52.30%	21.86%
Appendicitis	12.23%	16.83%	16.03%
Australian	34.89%	41.79%	22.97%
Balance	33.42%	38.02%	12.88%
Cleveland	67.10%	67.47%	51.75%
Dermatology	62.34%	61.46%	37.37%
Glass	50.16%	61.30%	49.16%
Hayes Roth	64.36%	63.46%	35.26%
Heart	31.20%	28.44%	17.80%
HouseVotes	6.13%	11.99%	3.67%
Ionosphere	16.22%	19.83%	10.33%
Liverdisorder	30.84%	36.97%	28.73%
Mammographic	21.38%	30.41%	17.25%
Parkinsons	17.42%	33.81%	17.37%
Pima	25.78%	27.83%	24.00%
Popfailures	7.04%	7.08%	5.44%
Regions2	38.29%	39.98%	25.81%
Ring	21.65%	50.36%	2.09%
Saheart	32.19%	33.90%	29.38%
Segment	59.68%	54.25%	39.44%
Sonar	27.85%	34.20%	19.62%
Spiral	44.87%	50.02%	18.98%
Tae	60.07%	61.78%	52.44%
Thyroid	10.52%	8.53%	7.12%
Wdbc	7.27%	8.82%	5.29%
Wine	31.41%	31.47%	8.67%
Z_F_S	13.16%	23.37%	4.21%
ZO_NF_S	9.02%	22.18%	4.17%
ZONF_S	4.03%	17.41%	2.18%
ZOO	21.93%	33.50%	9.00%

Table 4. Regression error for different datasets.

Dataset	**KRBF**	**GRBF**	**Proposed**
Abalone	2559.48	4161.66	1960.22
Airfoil	5.49	18.15	0.58
Anacalt	11.628	5.58	0.003
BK	0.17	0.21	0.23
BL	0.05	0.019	0.0009
Concrete	1.15	1.50	0.52

Table 4. *Cont.*

Dataset	KRBF	GRBF	Proposed
Housing	2884.09	4784.50	693.22
Laser	2.35	6.94	1.04
MB	11.33	2.44	0.63
NT	72.14	0.22	0.09
Quake	15.36	171.43	7.86

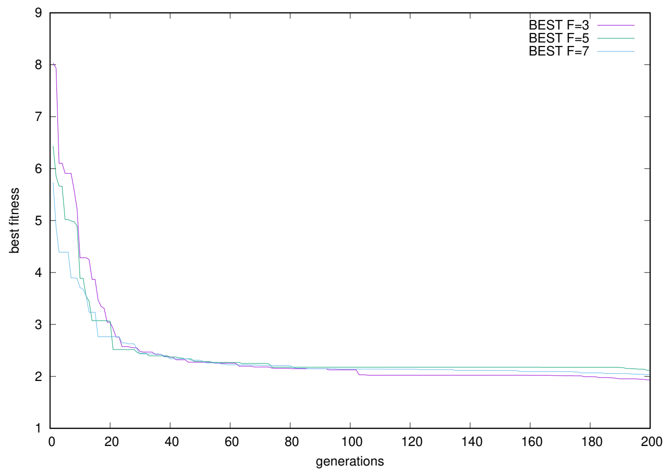

Figure 1. Plot of best fitness for the Wine problem for different values of parameter F.

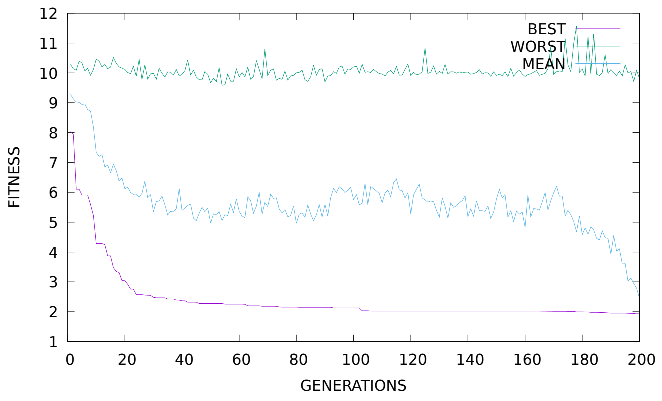

Figure 2. Plot of best, worst and average fitness for the Wine dataset and $F = 3$.

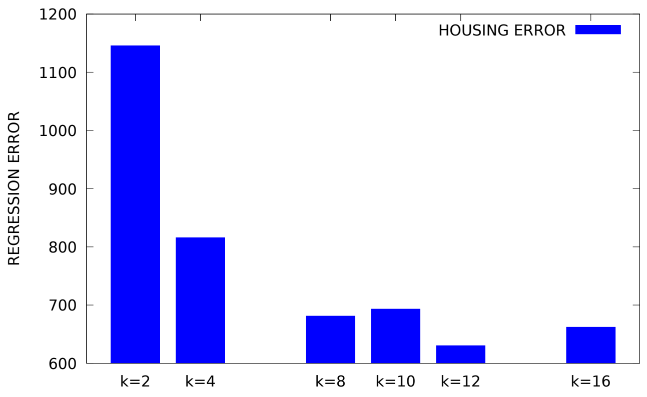

Figure 3. Experiments with different values of k for the Housing dataset.

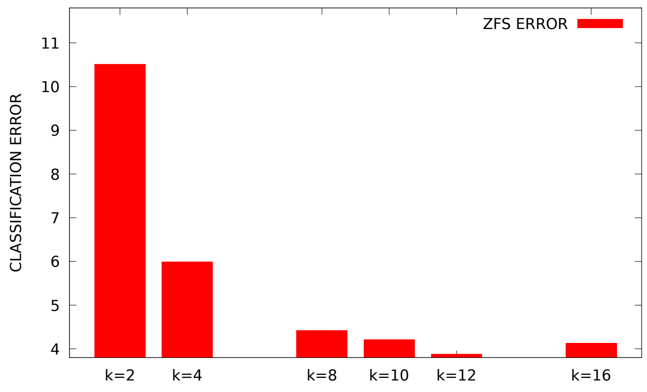

Figure 4. Classification error of Z_F_S dataset for different values of *k*.

Table 5. Comparison of precision and recall between the traditional RBF and the proposed method for some datasets.

Dataset	Precision KRBF	Recall KRBF	Precision Proposed	Recall Proposed
Housevotes	90.61%	95.60%	96.44%	94.25%
Spiral	55.53%	55.93%	82.93%	84.66%
ZONF_S	92.76%	87.23%	97.27%	94.55%

4. Conclusions

A two-phase method was proposed in this article to train RBF neural networks for classification and regression problems. Firstly, a commonly used clustering method was used to estimate an interval for the critical parameters of the neural network. Subsequently, a parallel genetic algorithm was incorporated to locate the best RBF network with good generalization capabilities. The used algorithm was coded using ANSI C++ and open source libraries such as the Armadillo library and the OpenMP library for parallelization. Future research may include:

1. Using parallel methods for the k-means clustering phase of the method.
2. Dynamic selection of k in k-means algorithm.
3. More advanced stopping rules for the genetic algorithm.
4. Replacing the genetic algorithm with other optimization methods such as particle swarm optimization, ant colony optimization, etc.

Author Contributions: I.G.T., A.T. and E.K. conceived of the idea and methodology and supervised the technical part regarding the software for the RBF trainig method. I.G.T. conducted the experiments, employing several datasets, and provided the comparative experiments. A.T. performed the statistical analysis. E.K. and all other authors prepared the manuscript. E.K. and I.G.T. organized the research team and A.T. supervised the project. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge support of this work from the project "Immersive Virtual, Augmented and Mixed Reality Center of Epirus" (MIS 5047221) which is implemented under the Action "Reinforcement of the Research and Innovation Infrastructure", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The experiments of this research work were performed at the high performance computing system established at Knowledge and Intelligent Computing Laboratory, Department of Informatics and Telecommunications, University of Ioannina, acquired with the project "Educational Laboratory equipment of TEI of Epirus" with MIS 5007094 funded by the Operational Programme "Epirus" 2014–2020, by ERDF and national funds.

Conflicts of Interest: All authors declare that they have no conflict of interest.

References

1. Park, J.; Sandberg, I.W. Universal Approximation Using Radial-Basis-Function Networks. *Neural Comput.* **1991**, *3*, 246–257. [CrossRef]
2. Bishop, C.M. Neural networks and their applications. *Rev. Sci. Instrum.* **1994**, *65*, 1803–1832. [CrossRef]
3. Teng, P. Machine-learning quantum mechanics: Solving quantum mechanics problems using radial basis function networks. *Phys. Rev. E* **2018**, *98*, 033305. [CrossRef]
4. 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. [CrossRef]
5. Alexandridis, A.; Chondrodima, E.; Efthimiou, E.; Papadakis, G.; Vallianatos, F.; Triantis, D. Large Earthquake Occurrence Estimation Based on Radial Basis Function Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 5443–5453. [CrossRef]
6. 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. [CrossRef]
7. Wang, Y.P.; Dang, J.W.; Li, Q.; Li, S. Multimodal medical image fusion using fuzzy radial basis function neural networks. In Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, Ningbo, China, 9–12 July 2017; pp. 778–782.
8. Mehrabi, S.; Maghsoudloo, M.; Arabalibeik, H.; Noormand, R.; Nozari, Y. Congestive heart failure, Chronic obstructive pulmonary disease, Clinical decision support system, Multilayer perceptron neural network and radial basis function neural network. *Expert Syst. Appl.* **2009**, *36*, 6956–6959. [CrossRef]
9. Veezhinathan, M.; Ramakrishnan, S. Detection of Obstructive Respiratory Abnormality Using Flow–Volume Spirometry and Radial Basis Function Neural Networks. *J. Med. Syst.* **2007**, *31*, 461. [CrossRef]
10. Mai-Duy, N.; Tran-Cong, T. Numerical solution of differential equations using multiquadric radial basis function networks. *Neural Netw.* **2001**, *14*, 185–199. [CrossRef]
11. Mai-Duy, N. Solving high order ordinary differential equations with radial basis function networks. *Int. J. Numer. Meth. Eng.* **2005**, *62*, 824–852. [CrossRef]
12. Wan, C.; Harrington, P.d.B. Self-Configuring Radial Basis Function Neural Networks for Chemical Pattern Recognition. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1049–1056. [CrossRef]
13. Yao, X.; Zhang, X.; Zhang, R.; Liu, M.; Hu, Z.; Fan, B. Prediction of enthalpy of alkanes by the use of radial basis function neural networks. *Comput. Chem.* **2001**, *25*, 475–482. [CrossRef]
14. Shahsavand, A.; Ahmadpour, A. Application of optimal RBF neural networks for optimization and characterization of porous materials. *Comput. Chem. Eng.* **2005**, *29*, 2134–2143. [CrossRef]
15. Momoh, J.A.; Reddy, S.S. Combined Economic and Emission Dispatch using Radial Basis Function. In Proceedings of the 2014 IEEE PES General Meeting | Conference & Exposition, National Harbor, MD, USA, 27–31 July 2014; pp. 1–5. [CrossRef]
16. Guo, J.-J.; Luh, P.B. Selecting input factors for clusters of Gaussian radial basis function networks to improve market clearing price prediction. *IEEE Trans. Power Syst.* **2003**, *18*, 665–672.
17. Falat, L.; Stanikova, Z.; Durisova, M.; Holkova, B.; Potkanova, T. Application of Neural Network Models in Modelling Economic Time Series with Non-constant Volatility. *Procedia Econ. Financ.* **2015**, *34*, 600–607. [CrossRef]
18. Laoudias, C.; Kemppi, P.; Panayiotou, C.G. Localization Using Radial Basis Function Networks and Signal Strength Fingerprints in WLAN. In Proceedings of the GLOBECOM 2009—2009 IEEE Global Telecommunications Conference, Honolulu, HI, USA, 30 November–4 December 2009; pp. 1–6.
19. Chen, S.; Mulgrew, B.; Grant, P.M. A clustering technique for digital communications channel equalization using radial basis function networks. *IEEE Trans. Neural Netw.* **1993**, *4*, 570–590. [CrossRef]
20. Jiang, H.; Yang, Y.; Shi, M. Chemometrics in Tandem with Hyperspectral Imaging for Detecting Authentication of Raw and Cooked Mutton Rolls. *Foods* **2021**, *10*, 2127. [CrossRef]
21. Li, X.; Jiang, H.; Jiang, X.; Shi, M. Identification of Geographical Origin of Chinese Chestnuts Using Hyperspectral Imaging with 1D-CNN Algorithm. *Agriculture* **2021**, *11*, 1274. [CrossRef]
22. Fath, A.H.; Madanifar, F.; Abbasi, M. Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems. *Petroleum* **2020**, *6*, 80–91. [CrossRef]
23. Deng, Y.; Zhou, X.; Shen, J.; Xiao, G.; Hong, H.; Lin, H.; Wu, F.; Liao, B.Q. New methods based on back propagation (BP) and radial basis function (RBF) artificial neural networks (ANNs) for predicting the occurrence of haloketones in tap water. *Sci. Total Environ.* **2021**, *772*, 145534. [CrossRef]

24. Aqdam, H.R.; Ettefagh, M.M.; Hassannejad, R. Health monitoring of mooring lines in floating structures using artificial neural networks. *Ocean Eng.* **2018**, *164*, 284–297. [CrossRef]
25. Arenas, M.G.; Parras-Gutierrez, E.; Rivas, V.M.; Castillo, P.A.; del Jesus, M.J.; Merelo, J.J. Parallelizing the Design of Radial Basis Function Neural Networks by Means of Evolutionary Meta-algorithms. In Proceedings of the Bio-Inspired Systems: Computational and Ambient Intelligence, IWANN 2009, Salamanca, Spain, 10–12 June 2009; Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5517.
26. Brandstetter, A.; Artusi, A. Radial Basis Function Networks GPU-Based Implementation. *IEEE Trans. Neural Netw.* **2008**, *19*, 2150–2154. [CrossRef]
27. Kuncheva, L.I. Initializing of an RBF network by a genetic algorithm. *Neurocomputing* **1997**, *14*, 273–288. [CrossRef]
28. Kubat, M. Decision trees can initialize radial-basis function networks. *IEEE Trans. Neural Netw.* **1998**, *9*, 813–821. [CrossRef]
29. Franco, D.G.B.; Steiner, M.T.A. New Strategies for Initialization and Training of Radial Basis Function Neural Networks. *IEEE Lat. Am. Trans.* **2017**, *15*, 1182–1188. [CrossRef]
30. Ricci, E.; Perfetti, R. Improved pruning strategy for radial basis function networks with dynamic decay adjustment. *Neurocomputing* **2006**, *69*, 1728–1732. [CrossRef]
31. Bortman, M.; Aladjem, M. A Growing and Pruning Method for Radial Basis Function Networks. *IEEE Trans. Neural Netw.* **2009**, *20*, 1039–1045. [CrossRef]
32. 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. [CrossRef] [PubMed]
33. Chen, J.Y.; Qin, Z.; Jia, J. A PSO-Based Subtractive Clustering Technique for Designing RBF Neural Networks. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, 1–6 June 2008; pp. 2047–2052.
34. Esmaeili, A.; Mozayani, N. Adjusting the parameters of radial basis function networks using Particle Swarm Optimization. In Proceedings of the 2009 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, Hong Kong, 11–13 May 2009; pp. 179–181.
35. O'Hora, B.; Perera, J.; Brabazon, A. Designing Radial Basis Function Networks for Classification Using Differential Evolution. In Proceedings of the 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, USA, 16–21 July 2006; pp. 2932–2937.
36. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Davis Davis, CA, USA, 27 December 1965–7 January 1966; Volume 1, pp. 281–297.
37. Kaelo, P.; Ali, M.M. Integrated crossover rules in real coded genetic algorithms. *Eur. J. Oper.* **2007**, *176*, 60–76. [CrossRef]
38. Tsoulos, I.G. Modifications of real code genetic algorithm for global optimization. *Appl. Math. Comput.* **2008**, *203*, 598–607. [CrossRef]
39. Sanderson, C.; Curtin, R. Armadillo: A template-based C++ library for linear algebra. *J. Open Source Softw.* **2016**, *1*, 26. [CrossRef]
40. Chandra, R.; Dagum, L.; Kohr, D.; Maydan, D.; McDonald, J.; Menon, R. *Parallel Programming in OpenMP*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2001.
41. 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.
42. Tzimourta, K.D.; Tsoulos, I.; Bilero, 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. [CrossRef]
43. 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.: San Francisco, CA, USA, 1991.
44. Quinlan, J.R. Simplifying Decision Trees. *Int. Man–Mach. Stud.* **1987**, *27*, 221–234. [CrossRef]
45. Shultz, T.; Mareschal, D.; Schmidt, W. Modeling Cognitive Development on Balance Scale Phenomena. *Mach. Learn.* **1994**, *16*, 59–88. [CrossRef]
46. Zhou, Z.H.; Jiang, Y. NeC4.5: Neural ensemble based C4.5. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 770–773. [CrossRef]
47. Setiono, R.; Leow, W.K. FERNN: An Algorithm for Fast Extraction of Rules from Neural Networks. *Appl. Intell.* **2000**, *12*, 15–25. [CrossRef]
48. 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.
49. Valentini, G.; Masulli, F. NEURObjects: An object-oriented library for neural network development. *Neurocomputing* **2002**, *48*, 623–646. [CrossRef]
50. Denoeux, T. A neural network classifier based on Dempster-Shafer theory. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2000**, *30*, 131–150. [CrossRef]
51. Hayes-Roth, B.; Hayes-Roth, F. Concept learning and the recognition and classification of exemplars. *J. Verbal Learn. Verbal Behav.* **1977**, *16*, 321–338. [CrossRef]
52. Kononenko, I.; Šimec, E.; Robnik-Šikonja, M. Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF. *Appl. Intell.* **1997**, *7*, 39–55. [CrossRef]

53. 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. [CrossRef] [PubMed]
54. Dy, J.G.; Brodley, C.E. Feature Selection for Unsupervised Learning. *J. Mach. Learn. Res.* **2004**, *5*, 845–889.
55. Perantonis, S.J.; Virvilis, V. Input Feature Extraction for Multilayered Perceptrons Using Supervised Principal Component Analysis. *Neural Process. Lett.* **1999**, *10*, 243–252. [CrossRef]
56. Garcke, J.; Griebel, M. Classification with sparse grids using simplicial basis functions. *Intell. Data Anal.* **2002**, *6*, 483–502. [CrossRef]
57. Elter, M.; Schulz-Wendtland, 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. [CrossRef]
58. 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. [CrossRef]
59. 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, Washington, DC, USA, 6–9 November 1988; pp. 261–265.
60. 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. [CrossRef]
61. Giannakeas, N.; Tsipouras, 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, Milan, Italy, 25–29 August 2015; pp. 3097–3100.
62. Breiman, L. *Bias, Variance and Arcing Classifiers*; Tec. Report 460; Statistics Department, University of California: Berkeley, CA, USA, 1996.
63. Hastie, T.; Tibshirani, R. Non-parametric logistic and proportional odds regression. *J. R. Stat. Soc. Ser. C (Appl. Stat.)* **1987**, *36*, 260–276. [CrossRef]
64. Dash, M.; Liu, H.; Scheuermann, P.; Tan, K.L. Fast hierarchical clustering and its validation. *Data Knowl. Eng.* **2003**, *44*, 109–138. [CrossRef]
65. 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. [CrossRef]
66. Lim, T.S.; Loh, W.Y. A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. *Mach. Learn.* **2000**, *40*, 203–228. [CrossRef]
67. Quinlan, J.R.; Compton, P.J.; Horn, K.A.; Lazurus, L. Inductive knowledge acquisition: A case study. In Proceedings of the Second Australian Conference on Applications of Expert Systems, Sydney, Australia, 14–16 May 1986.
68. 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. [CrossRef]
69. 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.* **2003**, *33*, 802–813. [CrossRef]
70. Zhong, P.; Fukushima, M. Regularized nonsmooth Newton method for multi-class support vector machines. *Optim. Methods Softw.* **2007**, *22*, 225–236. [CrossRef]
71. 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. [CrossRef]
72. Koivisto, M.; Sood, K. Exact Bayesian Structure Discovery in Bayesian Networks. *J. Mach. Learn. Res.* **2004**, *5*, 549–573.
73. 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*; Technical Report No. 48; Sea Fisheries Division-Department of primary Industry and Fisheries: Hobart, Australia, 1994; ISSN 1034-3288.
74. Brooks, T.F.; Pope, D.S.; Marcolini, A.M. *Airfoil Self-Noise and Prediction*; Technical Report, NASA RP-1218; NASA: Hampton, VA, USA, 1989.
75. Simonoff, J.S. *Analyzing Categorical Data*; Springer: New York, NY, USA, 2003.
76. Simonoff, J.S. *Smooting Methods in Statistics*; Springer: New York, NY, USA, 1996.
77. Yeh, I.C. Modeling of strength of high performance concrete using artificial neural networks. *Cem. Concr. Res.* **1998**, *28*, 1797–1808. [CrossRef]
78. Harrison, D.; Rubinfeld, D.L. Hedonic prices and the demand for clean air. *J. Environ. Econ. Manag.* **1978**, *5*, 81–102. [CrossRef]
79. 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. Am. Med. Assoc.* **1992**, *268*, 1578–1580 [CrossRef]
80. Ding, S.; Xu, L.; Su, C.; Jin, F. An optimizing method of RBF neural network based on genetic algorithm. *Neural Comput. Appl.* **2012**, *21*, 333–336. [CrossRef]