Computers, Materials & Continua
DOI: 10.32604/cmc.2023.032329
Article

Numerical Comparison of Shapeless Radial Basis Function Networks in Pattern Recognition

Sunisa Tavaen and Sayan Kaennakham*

School of Mathematics, Institute of Science, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand
*Corresponding Author: Sayan Kaennakham. Email: sayan_kk@g.sut.ac.th
Received: 14 May 2022; Accepted: 23 August 2022

Abstract: This work focuses on radial basis functions containing no parameters with the main objective being to comparatively explore more of their effectiveness. For this, a total of sixteen forms of shapeless radial basis functions are gathered and investigated under the context of the pattern recognition problem through the structure of radial basis function neural networks, with the use of the Representational Capability (RC) algorithm. Different sizes of datasets are disturbed with noise before being imported into the algorithm as 'training/testing' datasets. Each shapeless radial basis function is monitored carefully with effectiveness criteria including accuracy, condition number (of the interpolation matrix), CPU time, CPU-storage requirement, underfitting and overfitting aspects, and the number of centres being generated. For the sake of comparison, the well-known Multiquadric-radial basis function is included as a representative of shape-contained radial basis functions. The numerical results have revealed that some forms of shapeless radial basis functions show good potential and are even better than Multiquadric itself indicating strongly that the future use of radial basis function may no longer face the pain of choosing a proper shape when shapeless forms may be equally (or even better) effective.

Keywords: Shapeless RBF-neural networks; pattern recognition; large scattered data

1 Introduction

Under the framework of intelligent machines, the ability to classify or recognise patterns through the structure of neural networks is highly essential and challenging in a wide range of applications including biomedical and biology, social medial intelligence (SMI), video surveillance, intelligent retail environment and digital cultural heritage [1,2]. Several concepts or methodologies have been proposed and studied during the past decade where networks can be roughly categorised into two main classes; deep learning and shallow learning, the name represents the number of layers involved in the network. Some recent successes of deep networks to be recommended include the classification of Alzheimer's diseases and mild cognitive impairment [3] and the classification of COVID-19 based on CT images [4].

On the other hand, the shallow structure has also been comparatively studied and applied in a variety of contexts [5]. One of the popular shallow architectures is the use of the so-called 'radial basis function (RBF)', a type of multivariate function.

RBF neural networks were originally introduced in [6,7] and have been extensively investigated ever since such as the classification and prediction [8–10], those in agriculture [11,12], finance [13], and medicine [14], to mention only a few, confirming the versatile and promising aspect of RBF networks in general. When it comes to applying RBF networks, two traditional and well-known choices are the Gaussian (GA) and the Multiquadric (MQ), in which a so-called 'shape parameter' occurs. It is this shape parameter that is known to play a highly crucial role in determining the quality of the networks. Specifying a proper value of this shape is rather complicated. It is known to strongly depend on several factors including the number and the distribution manner of nodes, the size of the problem domain, the numerical and mathematical procedures involved in the process, etc. Apart from GA and MQ, the pain of shape-determining also persists for all shape-contained RBF types making their use not as practical. The challenge of searching for an optimal shape parameter (if any and for each shape-contained RBF) has consequently been one of the active research areas related to RBFs' applications and some are briefly introduced here.

In 2019, Krowiak et al. [15] proposed a scheme employing a heuristic that relates the condition number to computational precision and searches for the largest value of this parameter that still ensures stable computation using geometrical dependence. A year later, Zheng et al. [16] used the approximation error of the least-squares to measure the shape parameter of the radial basis function and proposed a corresponding optimization problem to obtain the optimal shape parameter. Later in 2021, Kaennakham et al. [17] compared two interesting forms of shape (one is in exponential base and the other in trigonometric) in the context of recovering functions and their derivatives, and also in this year, Cavoretto et al. [18] proposed finding an optimal value of the shape parameter by using leave-one-out cross-validation (LOOCV) technique combined with univariate global optimization methods. Despite the broad range of research under this path, the topic is still highly problem-dependent, i.e., one reasonable good shape for a certain task might be useless for others when problem configuration changes.

Alternatively to shape-contained RBFs and to avoid the demerits mentioned so far, RBFs containing no shapes, referred to as 'shapefree' or 'shapeless' have recently been paid attention to. Recent work can be found in [19] where the authors attempted to approximate the solution of PDEs using RBF with no shapes. The application in the pattern recognition context, to our knowledge, has not been explored as much and our preliminary investigations have strongly been promising when shapeless RBFs are in use [20,21]. Therefore, to encourage the research under this path and extend the scope of our previous works, this work focuses on a numerical investigation of the effectiveness of sixteen shapeless RBFs. The challenge is carried out under the context of pattern recognition through the structure of radial basis function neural networks (RBF-NNs) applied in conjunction with the Representational Capability (RC) algorithm (full detail is provided in Section 2). The effectiveness is validated based on several criteria including accuracy, condition number, CUP-time and storage, user interference, and sensitivity to other factors.

Three main contributions this work is aimed to make are;

- To demonstrate the application of the Representational Capability (RC) algorithm in RBF neural networks for pattern recognition problems.
- To provide more information on how reliable shapeless RBFs can be for pattern recognition problems, as compared to one of the popular choices of shape-contained RBFs.

- To compare the effectiveness of sixteen selected shapeless RBFs under the same problem configuration.

It is hoped that with a good choice of shapeless RBF, one should be able to achieve high-quality results without suffering the pain caused by choosing the shape. Section 2 provides the problem statement with all crucial components before the computational algorithm with all the criteria being declared in Section 3. Section 4 then reports the essential results obtained and the main conclusions are drawn in Section 5.

2 Methodology and Problem Statement

2.1 Pattern Recognition with Radial Basis Functions Neural Networks

For the task of pattern recognition, the so-called 'training datasets' undergo a process designed to construct a model or a mapping (if any) that is hoped to best represent the rest of the data or 'testing datasets'. The process occurring in the middle is the main challenge as it is typically unknown. Both sets of data can be of the following form.

$$D = \left\{(\mathbf{x}_i, y_i)|\mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}, i = 1, 2, \cdots, n\right\} \tag{1}$$

where $\mathbf{x}_i$ are inputs and y_i are the corresponding outputs. The main task is to find a mapping D from the $d-$dimensional input space to $1-$dimensional output space.

The model structure of a typical radial basis function network is depicted in Fig. 1, when a data set $\{(\mathrm{x}_i, y_i)\}_{i=1}^n$ is given and the model searches for the corresponding output estimate $\hat{y}$ for the input vector $\mathbf{x}$, represented by functional form:

$$\hat{y} = f\,(\mathrm{x}) = \sum_{j=1}^{m} w_j \phi_j\,(\mathrm{x}) = \sum_{j=1}^{m} w_j \phi_j\left(\left|\left|\mathrm{x} - \mu_j\right|\right|_2\right) \tag{2}$$

where $||-||_2$ is the Euclidean distance norm, $\phi(\cdot)$ is a basis function, m is the number of basis functions, n is the number of input data, μ_j are the centres of j-th basis functions, and w_j are the weights associated with the j-th basis function. (note that $m \leq n$)

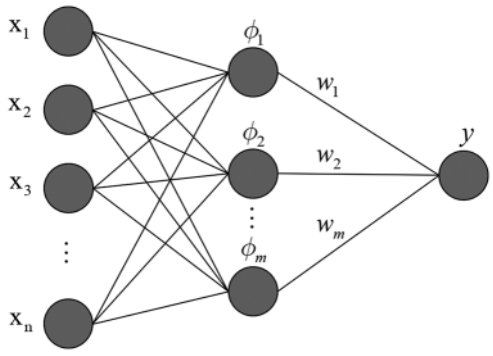

Figure 1: Typical radial basis function (RBF) neural network structure

2.2 The Selected Shapeless RBFs

Sixteen forms of shapeless RBFs are paid attention to in this work and they are listed in Table 1. For the sake of comparison, the well-known shape-contained RBF namely 'Multiquadric, MQ' is also parallelly studied where the shape choosing scheme proposed by Carlson, 1991 [22] is used. Differences in the curve profiles for each shapeless RBF can be seen and noticed in Fig. 2.

Table 1: Type of radial basis function

No.	Name of RBF	Abbreviation	Definition
1	Multiquadric	MQ	$\sqrt{r^2+\varepsilon^2}$
2	Polyharmonic spline	PS	$\begin{cases} r^{2k-1}, k \in \mathbb{N} \\ r^{2k}\ln(r), k \in \mathbb{N} \end{cases}$
3	Thin plate splines	TPS	$r^2 \log r$
4	Wu's CS-RBFs [23]	WU1	$(1-r)_+^7 \left(5r^6+35r^5+101r^4+147r^3+101r^2+35r+5\right)$
5	Wu's CS-RBFs [23]	WU2	$(1-r)_+^6 \left(6r^5+30r^4+72r^3+82r^2+36r+6\right)$
6	Wu's CS-RBFs [23]	WU3	$(1-r)_+^5 \left(5r^4+25r^3+48r^2+40r+8\right)$
7	Wu's CS-RBFs [23]	WU4	$(1-r)_+^4 \left(5r^3+20r^2+29r+16\right)$
8	Wendland's CS-RBFs [24]	WL1	$(1-r)_+$
9	Wendland's CS-RBFs [24]	WL2	$(1-r)_+^3 (3r+1)$
10	Wendland's CS-RBFs [24]	WL3	$(1-r)_+^5 \left(8r^2+5r+1\right)$
11	Wendland's CS-RBFs [24]	WL4	$(1-r)_+^2$
12	Wendland's CS-RBFs [24]	WL5	$(1-r)_+^4 (4r+1)$
13	Wendland's CS-RBFs [24]	WL6	$(1-r)_+^6 \left(35r^2+18r+3\right)$
14	Wendland's CS-RBFs [24]	WL7	$(1-r)_+^8 \left(32r^3+25r^2+8r+1\right)$
15	Buhmann [25]	BUH1	$\left(2r^4\log(r)-\frac{7r^4}{2}+\frac{16r^3}{3}-2r^2+\frac{1}{6}\right)_+$
16	Buhmann [25]	BUH2	$\left(\frac{112r^{\frac{9}{2}}}{45}+\frac{16r^{\frac{7}{2}}}{3}-7r^4-\frac{14r^2}{15}+\frac{1}{9}\right)_+$
17	Buhmann [25]	BUH3	$\left(\frac{1}{18}-r^2+\frac{4r^3}{9}+\frac{r^4}{2}-\frac{4r^3\log(r)}{3}\right)_+$

2.3 The Representational Capability (RC) Algorithm

The Representational Capability (RC) algorithm as proposed by Shin et al. (2000) [26] is a method to drag out only meaningful and significant information from the interpolation matrix when RBF is in use. For a given set of input data $\{x_i, y_i\}_{i=1}^n$, for the sake of the completeness of the work, the algorithm is elaborated with the following steps.

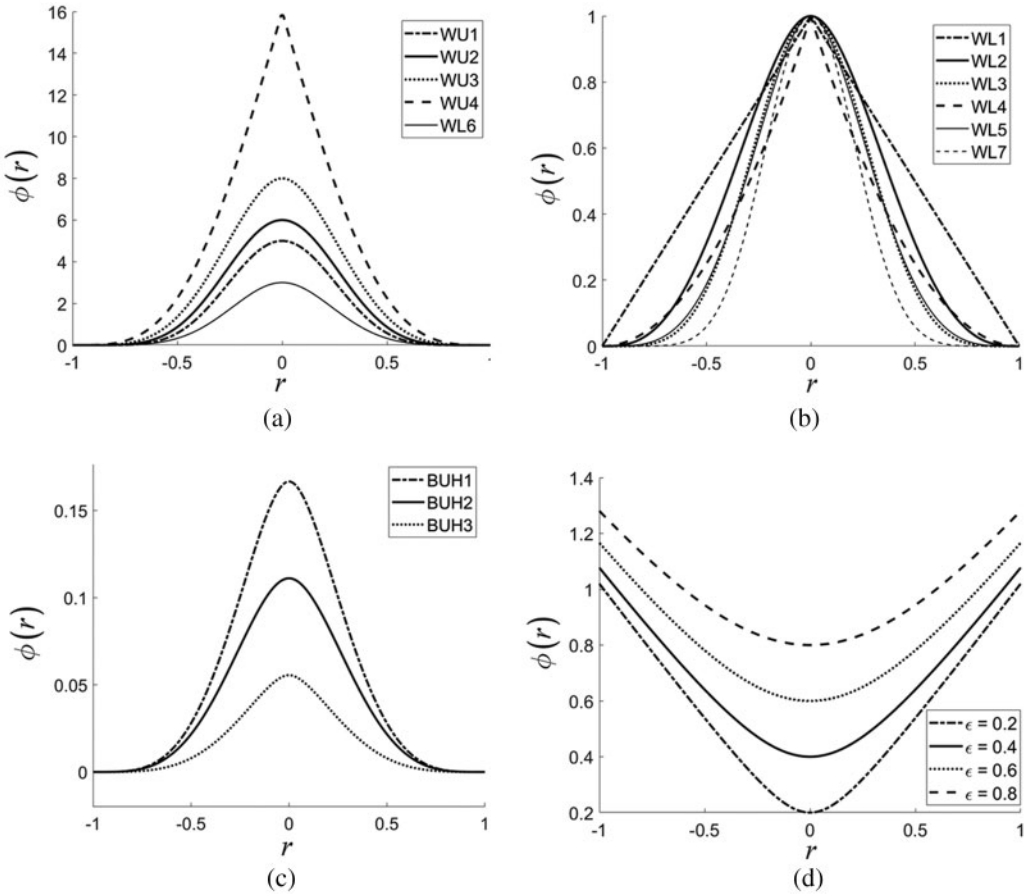

Figure 2: One-dimensional curve for each type of RBFs under this investigation; (a–c) of those containing no shapes (or shapeless), and (d) the famous MQ with different shape values

Step 1: Select a value for the width (or the shape parameter) (ε), and the effect of noise (δ) which δ is usually taken to be 0.1% to 1.0%, then construct the interpolation matrix G. For example, if the Gaussian basis function is used, so that

$$G = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix} \tag{3}$$

where $g_{ij} = \exp\left(-\left|\left|\mathbf{x}_i - \mathbf{x}_j\right|\right|^2 / 2\varepsilon^2\right)$ for $i, j = 1, 2, \ldots, n$.

Step 2: Determine the number of basis functions (m) by applying the singular value decomposition procedure to the interpolation matrix G. This yields a diagonal matrix (namely V) of singular values $s_1 \geq s_2 \geq \cdots \geq s_n \geq 0$. From this step, m can be determined from the following expression:

$$m = \max_{0 \leq i \leq n} \left\{ i | s_{i+1} \leq s_1 \times \frac{\delta}{100} \right\}. \tag{4}$$

Step 3: Determine the centres of basis function (μ). This is done by partitioning matrix V from singular value decomposition of the interpolation matrix G, as follows:

$$V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}_{n \times n} \tag{5}$$

Next, generate matrix $V' = \begin{bmatrix} v_{11}^T & v_{21}^T \end{bmatrix}_{m \times n}$ and apply QR factorization with column pivoting of matrix V'. Then compute $X^T P$ and choose the first m elements in $X^T P$ be the centres of basis function which are:

$$\mu = \left\{ \mu_j \right\}_{j=1}^{m} \tag{6}$$

Step 4: Compute the weight parameters from the basis function (w). Consider,

$$\boldsymbol{\phi} = \left[\phi_{i,j} \right], \text{For } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, m. \tag{7}$$

Compute the m weights using the following expression.

$$w = \boldsymbol{\phi}^{+} y \tag{8}$$

where $\boldsymbol{\phi}^{+}$ denoted the pseudo inverse of $\boldsymbol{\phi}$.

3 Experimental Setups

3.1 Performance Criteria

As previously mentioned, all sixteen shapeless RBFs are comparatively and numerically investigated. Therefore, proper and all-around effectiveness criteria are required and they are listed below.

- *Accuracy:* This process is carried out using the following two error norms.

$$L_{RMSE} = \left[\frac{1}{N} \sum_{i=1}^{N} \left(y^{ext.}(x_i, y_i) - y^{appx.}(x_i, y_i) \right)^2 \right]^{1/2} \tag{9}$$

and

$$L_{\infty} = \max_{1 \le i \le N} \left| y^{ext.}(x_i, y_i) - y^{appx.}(x_i, y_i) \right|. \tag{10}$$

- *Condition number:* The system can be solvable if the collocation matrix, $\boldsymbol{\varphi}$, has an inverse and this can be indicated by the means of its condition number ($Cond_{\delta}(-)$) expressed as follows.

$$Cond_{\delta}(\boldsymbol{\varphi}) = ||\boldsymbol{\varphi}||_{\delta} \left|\left| \boldsymbol{\varphi}^{-1} \right|\right|_{\delta} \tag{11}$$

The trending behaviour of this number is also recorded throughout this experiment, providing information on the solvability, [27], of the collocation matrix for future uses.

- *CPU-time:* With less amount of time required for the computational process, a method would be more desirable. The 'tic-toc' command in MATLAB is employed for this task.
- *Storage requirement:* Each computation step involved in the algorithm should ideally take as least amount of storage space as possible. For the sake of fairness, all experiments are carried out on the same computer (Intel(R) Core(TM) i7-1065G7 CPU @ 1.30 GHz 1.50 GHz with RAM 16.00 GB and 64-bit Operating System).

- *User's Interference:* A good algorithm should be able to process entirely by itself, no human interruption or judgments should be involved. This is all for future practical uses.
- *Sensitivity to parameters:* A small change in parameters embedded in the process should have as little effect as possible on the overall performance. Methods containing no parameters would be best in practice.
- *Ease of implementation:* Once the desirable method has been proven with small test models, it is then anticipated to successfully be implemented to larger problems with no difficulties (in terms of both mathematical structures and programming/coding).

3.2 Dataset-Partitioning Manners

For the data-partitioning process, the suggestion provided by Lazzaro et al. (2002) [28] is followed here with the information described as follows.

3.2.1 The Training Datasets

Two datasets with the size of 50×50 nodes are generated within a $[0, 1] \times [0, 1]$ domain; uniformly and randomly as shown in Fig. 3. These sets of data are to be used as a 'Training Dataset' for all numerical experiments in this study.

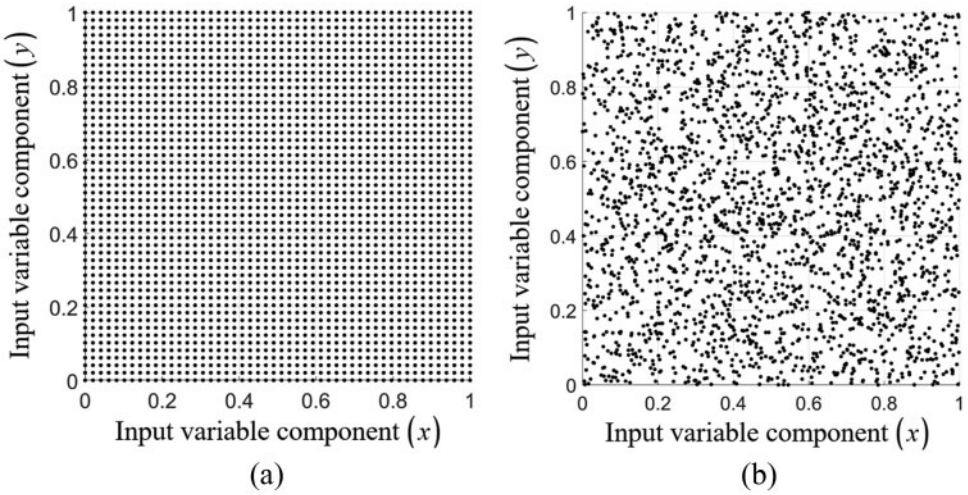

Figure 3: $(x, y)-$Node distribution manners for training datasets using 50×50 nodes: (a) uniformly (Ufm.), and (b) non-uniformly (randomly, Rdm.)

3.2.2 The Testing Datasets

To monitor and record the effectiveness of each RBF type, three large datasets are generated within the same domain in both manners uniformly (Ufm.) and randomly (Rdm.), and they contain 10000, 20164, and 30276 nodes [28].

4 Numerical Experiments, Results, and General Discussion

In this section, two well-known benchmarking test cases are numerically experimented with using all the RBF forms mentioned above. Nevertheless, with the limitation of the space, only those significant and relevant results are being mentioned, illustrated, and discussed.

4.1 Experiment 1

This first experiment is concerned with one of the most well-known testing functions invented by Franke in 1982 [29] and is to be referred to as 'Franke's function' hereafter. The expression of the function is as follows:

$$z = f(x,y) = 0.75\exp\left(-\frac{(9x-2)^2+(9y-2)^2}{4}\right) + 0.75\exp\left(-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right)$$
$$+ 0.5\exp\left(-\frac{(9x-7)^2+(9y-3)^2}{4}\right) - 0.2\exp\left(-(9x-4)^2-(9y-7)^2\right) \quad (12)$$

4.1.1 Performance at SNR = 30

The results validation for all RBFs for this first case is to be numerically investigated at two different noise levels (*SNR*); *SNR* = 15 and 30, as illustrated in Figs. 4 and 7.

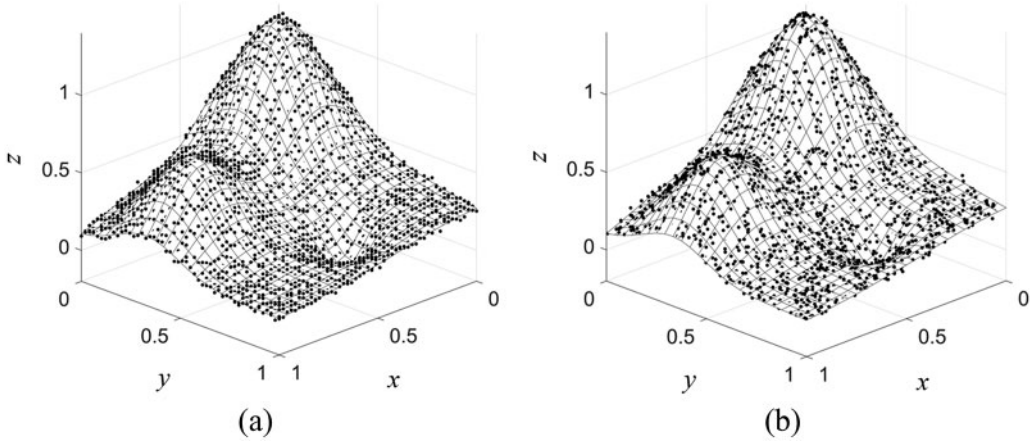

Figure 4: Noised-$z(x,y)$ training datasets with $SNR = 30$ based on two manners of distribution of $\{(x,y)\}$: (a) uniformly and (b) non-uniformly (randomly)

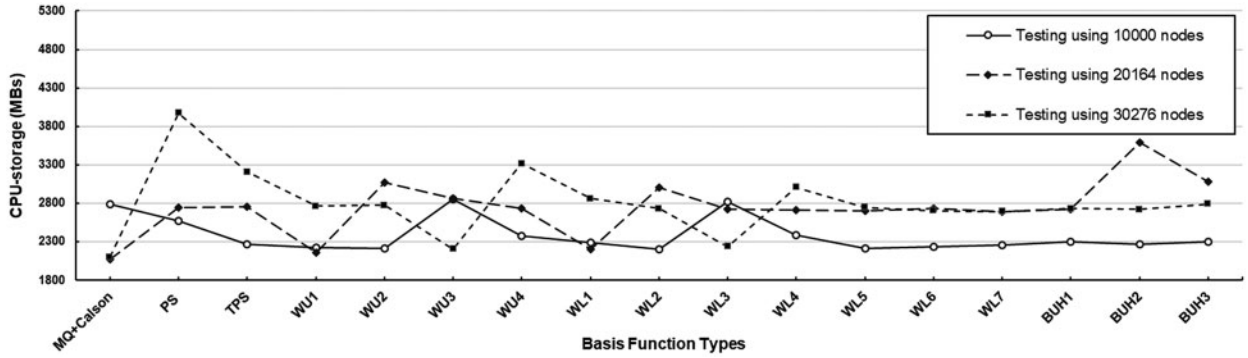

Figure 5: CPU-storage measurement observed at three different sizes of testing datasets with uniform node distribution using $SNR = 30$

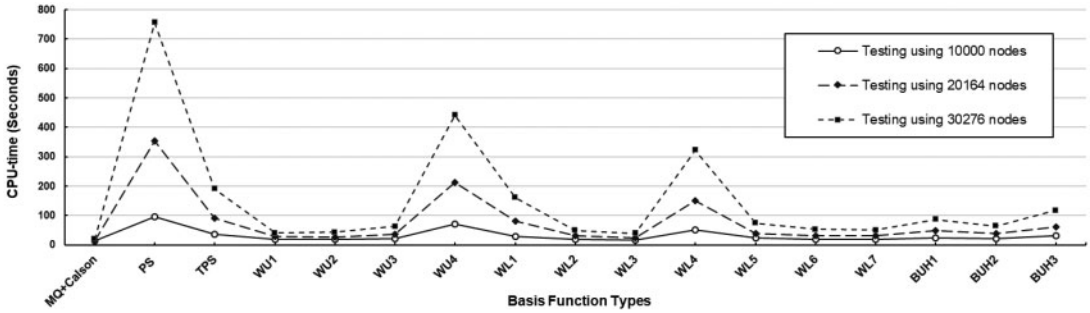

Figure 6: CPU-time measurement observed at three different sizes of testing datasets as uniform node distribution using $SNR = 30$

4.1.2 Performance at SNR = 15

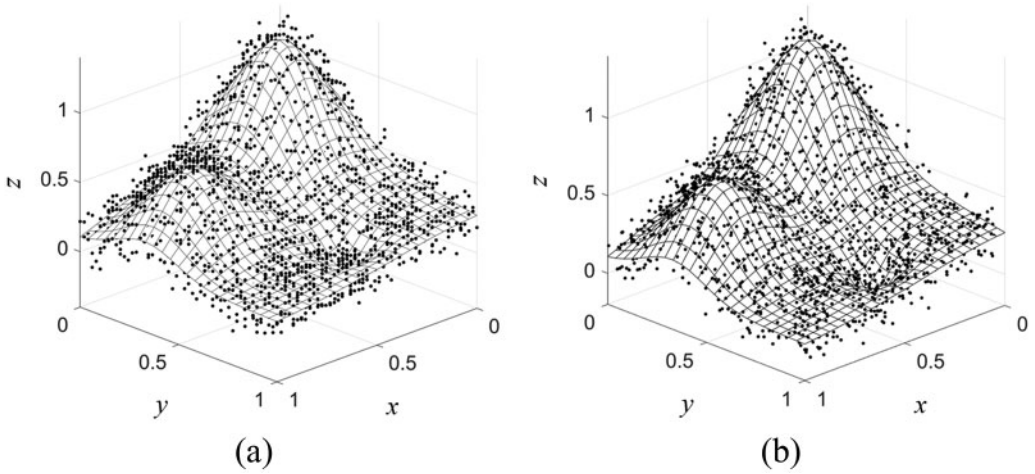

Figure 7: Noised-$z(x, y)$ training datasets with $SNR = 15$ based on two manners of distribution of $\{(x, y)\}$: (a) uniformly and (b) non-uniformly (randomly)

As can be seen in Figs. 4 and 7, the accuracy measured by both L_∞ and L_{RMSE} from both cases of node distribution manners slightly decreases with the increase of noise variance. This is not surprising as the pattern becomes more complicated with data being more disturbed by noise.

In terms of the number of basis functions (m) generated by the RC-algorithm, at $SNR = 30$ and with comparatively low values of L_∞, L_{RMSE}, and $Cond_\delta(-)$, it is found that $m = 87$, $m = 100$ and $m = 83$ are reproduced for WU3, WL5, and BUH2, respectively. This figure can be seen as a good success of these RBFs, considering 2,500 original nodes, while good accuracy is still achievable (see Tabls. 2 and 4). Table 3 also shows a similar figure for these RBFs at $SNR = 30$ confirming their potential for further uses. The ability to preserve the data's pattern while using a small number of nodes is desirable for any kind of RBF in practical applications. The small number in m also strongly affects both CPU time and storage required for each computation process as clearly seen in Figs. 5 and 6 for $SNR = 30$ and Figs. 8 and 9 for $SNR = 15$.

Table 2: Training and testing errors measured for Franke's function at $SNR = 30$

Types of RBF	Uniform node distribution				Random node distribution			
	L_∞		L_{RMSE}		L_∞		L_{RMSE}	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
MQ + Carlson	1.77E-01	1.55E-01	3.57E-02	3.18E-02	1.99E-01	2.19E-01	3.78E-02	3.58E-02
PS	5.50E-02	7.60E-01	1.15E-02	8.39E-02	5.75E-02	2.60E+00	1.24E-02	1.20E-01
TPS	5.48E-02	3.54E-02	1.51E-02	6.50E-03	5.73E-02	1.52E-01	1.49E-02	7.60E-03
WU1	1.07E-01	8.19E-02	2.04E-02	1.33E-02	9.10E-02	8.66E-02	2.08E-02	1.45E-02
WU2	8.03E-02	5.80E-02	1.84E-02	1.00E-02	6.03E-02	5.00E-02	1.72E-02	8.20E-03
WU3	5.80E-02	2.58E-02	1.57E-02	3.90E-03	5.75E-02	2.26E-02	1.54E-02	3.80E-03
WU4	5.65E-02	3.07E-02	1.34E-02	7.10E-03	4.63E-02	3.08E-02	1.31E-02	7.10E-03
WL1	6.26E-02	5.13E-02	1.53E-02	6.00E-03	6.72E-02	7.88E-02	1.53E-02	6.60E-03
WL2	6.19E-02	3.72E-02	1.65E-02	6.20E-03	5.84E-02	3.09E-02	1.58E-02	4.90E-03
WL3	6.84E-02	4.64E-02	1.70E-02	7.50E-03	5.92E-02	3.61E-02	1.62E-02	5.40E-03
WL4	5.93E-02	2.98E-02	1.38E-02	6.70E-03	5.70E-02	3.90E-02	1.37E-02	6.30E-03
WL5	5.64E-02	1.76E-02	1.55E-02	3.70E-03	5.74E-02	2.01E-02	1.54E-02	3.90E-03
WL6	5.83E-02	2.45E-02	1.59E-02	4.40E-03	6.09E-02	3.71E-02	1.60E-02	5.70E-03
WL7	6.14E-02	3.72E-02	1.63E-02	5.80E-03	6.12E-02	3.74E-02	1.60E-02	5.60E-03
BUH1	5.74E-02	2.00E-02	1.55E-02	4.00E-03	5.56E-02	2.78E-02	1.52E-02	3.70E-03
BUH2	5.66E-02	1.95E-02	1.56E-02	3.60E-03	5.74E-02	2.39E-02	1.54E-02	3.50E-03
BUH3	5.68E-02	1.45E-02	1.51E-02	4.10E-03	5.63E-02	1.80E-02	1.50E-02	4.30E-03

Table 3: Training and testing errors measured for Franke's function at $SNR = 15$

Types of RBF	Uniform node distribution				Random node distribution			
	L_∞		L_{RMSE}		L_∞		L_{RMSE}	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
MQ + Carlson	3.50E-01	2.55E-01	9.84E-02	3.88E-02	4.02E-01	2.02E-01	9.30E-02	3.75E-02
PS	2.90E-01	4.27E+00	6.36E-02	3.28E-01	3.02E-01	3.92E+00	6.55E-02	3.34E-01
TPS	2.76E-01	1.56E-01	8.40E-02	3.18E-02	2.90E-01	1.92E-01	8.15E-02	3.19E-02
WU1	3.56E-01	8.46E-02	8.97E-02	1.76E-02	3.09E-01	1.06E-01	8.63E-02	1.91E-02
WU2	3.39E-01	6.11E-02	8.93E-02	1.56E-02	3.00E-01	7.43E-02	8.54E-02	1.55E-02
WU3	3.26E-01	6.36E-02	8.82E-02	1.81E-02	2.75E-01	6.31E-02	8.44E-02	1.82E-02
WU4	3.36E-01	2.02E-01	7.60E-02	4.27E-02	2.62E-01	1.92E-01	7.35E-02	3.82E-02
WL1	3.17E-01	1.03E-01	8.51E-02	2.91E-02	3.02E-01	1.11E-01	8.22E-02	2.53E-02
WL2	3.30E-01	6.13E-02	8.89E-02	1.66E-02	2.99E-01	6.40E-02	8.51E-02	1.57E-02
WL3	3.44E-01	5.93E-02	8.93E-02	1.49E-02	2.91E-01	6.91E-02	8.52E-02	1.52E-02
WL4	3.52E-01	1.68E-01	7.94E-02	3.87E-02	2.70E-01	1.61E-01	7.66E-02	3.42E-02
WL5	3.22E-01	6.49E-02	8.81E-02	1.86E-02	2.72E-01	5.87E-02	8.44E-02	1.83E-02
WL6	3.32E-01	7.15E-02	8.87E-02	1.64E-02	2.77E-01	7.17E-02	8.49E-02	1.73E-02
WL7	3.29E-01	7.20E-02	8.89E-02	1.63E-02	2.83E-01	7.05E-02	8.50E-02	1.68E-02
BUH1	3.25E-01	7.75E-02	8.77E-02	2.04E-02	2.77E-01	8.06E-02	8.39E-02	2.06E-02

(Continued)

Table 3: Continued

Types of RBF	Uniform node distribution				Random node distribution			
	L_∞		L_{RMSE}		L_∞		L_{RMSE}	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
BUH2	3.20E-01	8.29E-02	8.83E-02	1.77E-02	2.73E-01	5.88E-02	8.46E-02	1.73E-02
BUH3	3.20E-01	1.03E-01	8.68E-02	2.34E-02	2.85E-01	9.31E-02	8.34E-02	2.25E-02

Table 4: The number of basis functions (m) and the condition number ($Cond_\infty(-)$) of all radial basis function candidates for the Franke's function test case

Types of RBF	$SNR = 30$				$SNR = 15$			
	m		$Cond_\infty(-)$		m		$Cond_\infty(-)$	
	Ufm.	Rdm.	Ufm.	Rdm.	Ufm.	Rdm.	Ufm.	Rdm.
MQ + Carlson	19	19	5.41E+03	6.44E+03	19	19	5.84E+03	6.02E+03
PS	1239	996	2.92E+04	2.39E+04	1239	996	2.92E+04	2.39E+04
TPS	295	243	1.40E+04	1.32E+04	295	243	1.40E+04	1.32E+04
WU1	41	39	8.26E+03	6.62E+03	41	39	8.26E+03	6.62E+03
WU2	47	47	6.99E+03	9.16E+03	47	47	6.99E+03	9.16E+03
WU3	87	83	9.01E+03	9.35E+03	87	83	9.01E+03	9.35E+03
WU4	689	662	9.29E+03	8.32E+03	689	662	9.29E+03	8.32E+03
WL1	245	234	9.62E+03	9.77E+03	245	234	9.62E+03	9.77E+03
WL2	64	62	9.28E+03	8.53E+03	64	62	9.28E+03	8.53E+03
WL3	51	50	7.02E+03	7.86E+03	51	50	7.02E+03	7.86E+03
WL4	511	504	9.25E+03	7.95E+03	511	504	9.25E+03	7.95E+03
WL5	100	94	1.23E+04	1.01E+04	100	94	1.23E+04	1.01E+04
WL6	69	65	1.07E+04	8.81E+03	69	65	1.07E+04	8.81E+03
WL7	64	62	8.71E+03	8.57E+03	64	62	8.71E+03	8.57E+03
BUH1	117	113	1.13E+04	1.01E+04	117	113	1.13E+04	1.01E+04
BUH2	83	78	1.22E+04	9.22E+03	83	78	1.22E+04	9.22E+03
BUH3	164	154	1.10E+04	1.24E+04	164	154	1.10E+04	1.24E+04

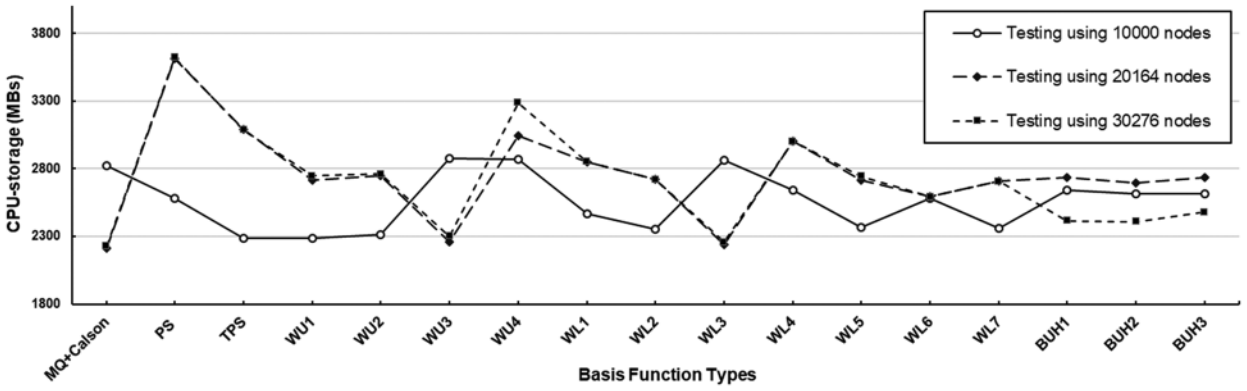

Figure 8: CPU-storage measurement observed at three different sizes of testing datasets with random node distribution using $SNR = 15$

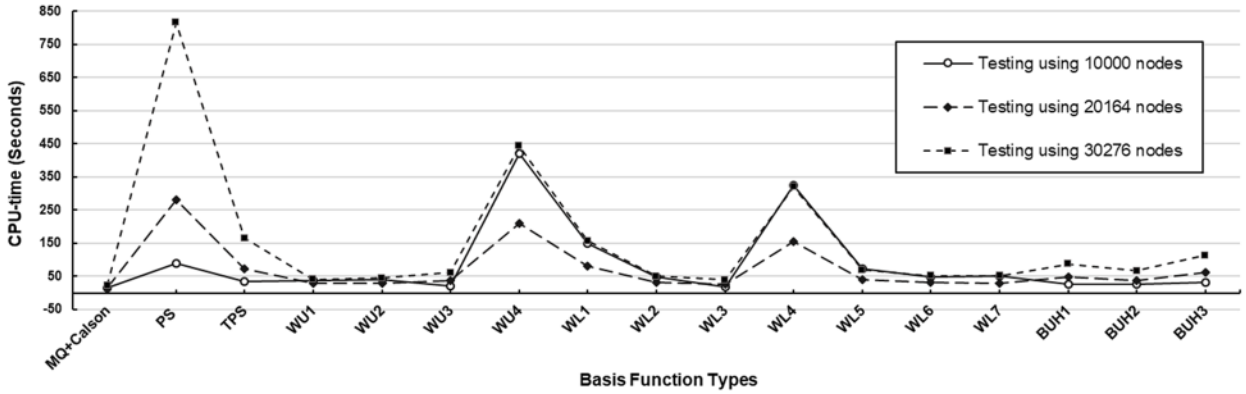

Figure 9: CPU-time measurement observed at three different sizes of testing datasets with random node distribution using $SNR = 15$

4.2 Experiment 2

For the second test case, we study one of the functions called F7 in the investigation nicely carried out by Renka et al. [30], (and shall be referred to as '**F7**' in this work as well). The function is defined as follows.

$$z = f(x, y) = 2\cos(10x)\sin(10y) + \sin(10xy) \tag{13}$$

Similar to the first experiment, two different noise levels or SNR ($SNR = 15$ and 30) are under investigation with their node distributions are illustrated in Figs. 10 and 13, respectively.

4.2.1 Performance at SNR = 30

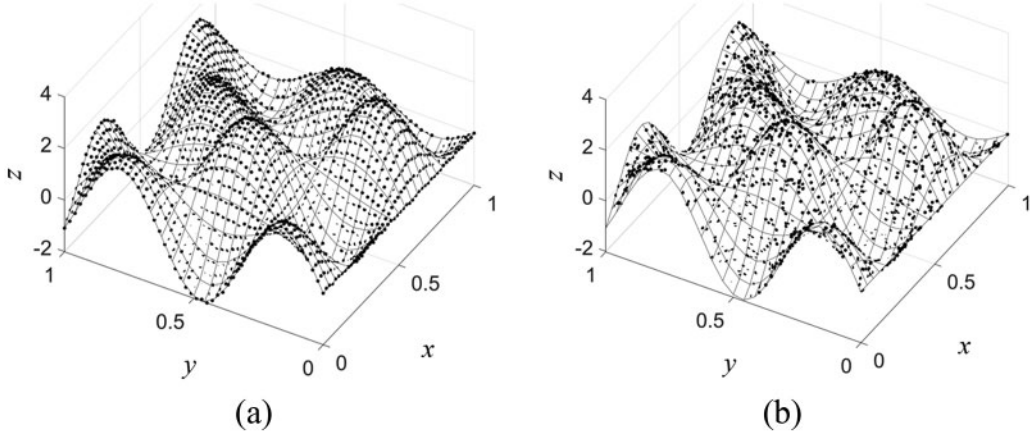

Figure 10: Noised-$z(x, y)$ training datasets based on two manners of distribution of $\{(x, y)\}$: (a) uniformly and (b) non-uniformly (randomly)

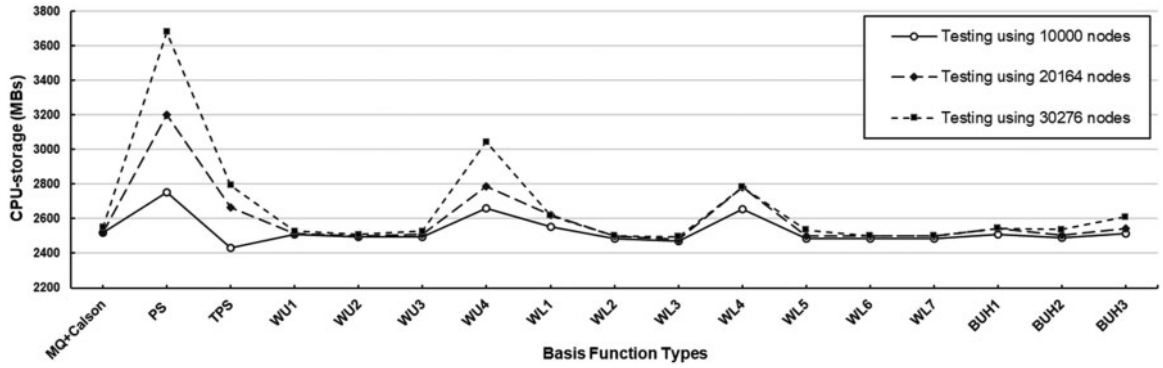

Figure 11: CPU-storage measurement observed at three different sizes of testing datasets with uniform node distribution using $SNR = 30$

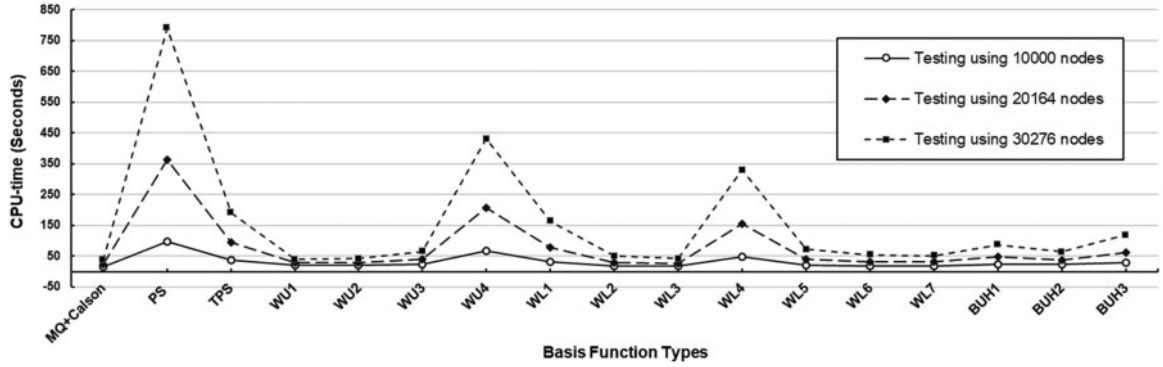

Figure 12: CPU-time measurement observed at three different sizes of testing datasets with uniform node distribution using $SNR = 30$

4.2.2 Performance at SNR = 15

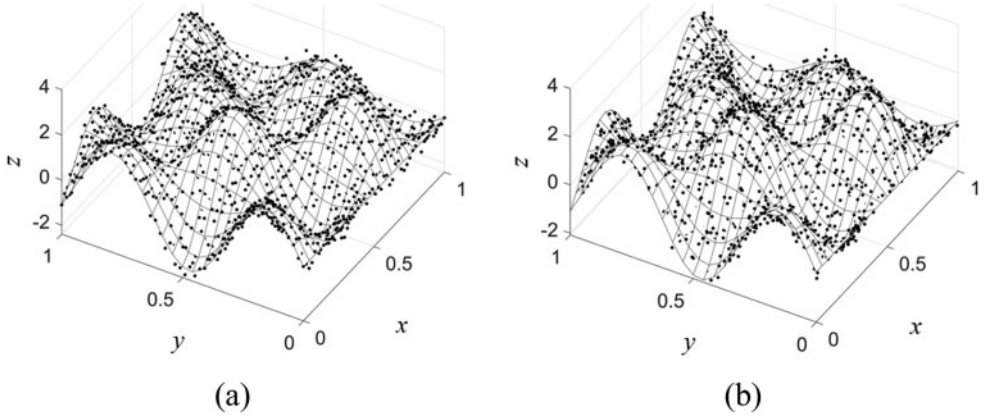

Figure 13: Noised-$z(x, y)$ training datasets based on two manners of distribution of $\{(x, y)\}$: (a) uniformly and (b) non-uniformly (randomly)

With the results obtained in these cases, shown in Tables 5–7, together with information provided in Figs. 11 and 12, it has been observed that the overall trends in accuracy (both L_∞ and L_{RMSE}) are similar to those discovered in the first experiment (with F1) where some smaller errors are noticed in the cases of WL6 and WL7. Moreover, the similarity to the first experiment in terms of the number of basis functions (m) produced by each shapeless RBF after undergoing the RC algorithm is also noticeable here. This is since with no parameter involved these values are dependent only on the distance between two nodes. In the case of MQ, nevertheless, with different values of shape parameters generated by the Carlson algorithm, the change in the number of basis functions (m) shall be anticipated.

Table 5: Training and testing errors measured for **F7** at $SNR = 30$

Types of RBF	Uniform node distribution				Random node distribution			
	L_∞		L_{RMSE}		L_∞		L_{RMSE}	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
MQ + Carlson	4.93E-01	4.38E-01	1.03E-01	9.38E-02	5.84E-01	6.33E-01	1.06E-01	1.10E-01
PS	2.90E-01	1.26E+01	6.65E-02	1.15E +00	2.33E-01	2.28E+01	4.78E-02	1.18E+00
TPS	3.88E-01	4.48E-01	6.88E-02	5.98E-02	4.55E-01	1.14E+00	6.24E-02	8.27E-02
WU1	7.16E-01	6.96E-01	1.49E-01	1.40E-01	1.12E+00	1.78E+00	1.71E-01	1.83E-01
WU2	4.84E-01	4.74E-01	9.45E-02	8.42E-02	5.00E-01	5.72E-01	8.64E-02	8.66E-02
WU3	1.93E-01	1.94E-01	4.65E-02	3.06E-02	2.14E-01	2.72E-01	4.46E-02	3.50E-02
WU4	1.14E-01	7.67E-02	3.03E-02	1.63E-02	1.39E-01	7.50E-01	2.94E-02	2.67E-02
WL1	1.08E+00	1.08E+00	7.74E-02	6.27E-02	8.93E-01	2.80E+00	7.54E-02	9.60E-02
WL2	3.99E-01	4.03E-01	7.21E-02	6.05E-02	6.38E-01	7.21E-01	7.25E-02	7.09E-02
WL3	4.92E-01	4.77E-01	9.19E-02	8.18E-02	2.31E-01	3.27E-01	5.80E-02	5.27E-02
WL4	1.29E-01	7.42E-02	3.20E-02	1.54E-02	1.69E-01	8.44E-01	3.20E-02	3.03E-02
WL5	1.86E-01	1.43E-01	3.94E-02	2.08E-02	1.98E-01	2.62E-01	4.20E-02	3.20E-02
WL6	2.00E-01	2.17E-01	4.17E-02	2.45E-02	2.18E-01	2.22E-01	4.33E-02	3.14E-02
WL7	1.89E-01	1.72E-01	4.41E-02	2.68E-02	1.95E-01	2.13E-01	4.23E-02	2.99E-02
BUH1	1.27E-01	8.67E-02	3.62E-02	1.60E-02	1.89E-01	5.10E-01	3.85E-02	2.76E-02
BUH2	1.66E-01	1.36E-01	4.10E-02	2.36E-02	2.17E-01	2.37E-01	4.55E-02	3.57E-02
BUH3	1.47E-01	8.09E-02	3.45E-02	1.30E-02	1.23E-01	4.18E-01	3.47E-02	1.88E-02

Table 6: Training and testing errors measured for **F7** at $SNR = 15$

Types of RBF	Uniform node distribution				Random node distribution			
	L_∞		L_{RMSE}		L_∞		L_{RMSE}	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
MQ + Carlson	1.12E+00	7.32E-01	2.37E-01	1.38E-01	9.91E-01	7.55E-01	2.15E-01	1.16E-01
PS	6.88E-01	1.58E+01	1.52E-01	1.35E+00	6.82E-01	2.56E+01	1.50E-01	1.35E+00
TPS	7.51E-01	4.71E-01	1.92E-01	8.70E-02	7.27E-01	1.28E+00	1.88E-01	1.10E-01
WU 1	9.39E-01	7.39E-01	2.40E-01	1.41E-01	1.26E+00	1.96E+00	2.54E-01	1.86E-01
WU 2	8.34E-01	5.10E-01	2.10E-01	8.69E-02	9.69E-01	5.20E-01	2.05E-01	8.92E-02
WU 3	7.31E-01	1.91E-01	1.93E-01	4.60E-02	7.85E-01	2.83E-01	1.90E-01	4.88E-02
WU 4	7.00E-01	4.72E-01	1.65E-01	8.91E-02	6.37E-01	8.65E-01	1.64E-01	8.95E-02
WL 1	1.15E+00	1.13E+00	1.96E-01	8.31E-02	8.27E-01	3.24E+00	1.93E-01	1.15E-01
WL 2	7.43E-01	3.51E-01	2.01E-01	6.72E-02	9.69E-01	6.64E-01	2.02E-01	7.77E-02
WL 3	8.41E-01	4.51E-01	2.10E-01	8.38E-02	9.16E-01	3.31E-01	1.94E-01	5.86E-02
WL 4	7.23E-01	3.70E-01	1.72E-01	8.04E-02	6.43E-01	8.41E-01	1.71E-01	7.98E-02

(Continued)

Table 6: Continued

Types of RBF	Uniform node distribution				Random node distribution			
	L_∞		L_{RMSE}		L_∞		L_{RMSE}	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
WL 5	7.08E-01	1.51E-01	1.90E-01	4.33E-02	8.01E-01	2.79E-01	1.89E-01	5.04E-02
WL 6	7.05E-01	1.57E-01	1.92E-01	3.88E-02	7.92E-01	2.69E-01	1.90E-01	4.36E-02
WL 7	7.05E-01	1.89E-01	1.92E-01	3.85E-02	7.88E-01	2.06E-01	1.90E-01	4.24E-02
BUH 1	7.05E-01	1.55E-01	1.89E-01	4.40E-02	7.40E-01	4.54E-01	1.87E-01	4.98E-02
BUH 2	7.14E-01	1.67E-01	1.91E-01	4.16E-02	8.34E-01	2.82E-01	1.91E-01	4.96E-02
BUH 3	7.20E-01	2.23E-01	1.87E-01	5.05E-02	7.82E-01	3.79E-01	1.84E-01	5.21E-02

Table 7: The number of basis functions (m) and the condition number ($Cond_\infty(-)$) of all radial basis function candidates for the **F7** test case

Types of RBF	$SNR = 30$				$SNR = 15$			
	m		$Cond_\infty(-)$		m		$Cond_\infty(-)$	
	Ufm.	Rdm.	Ufm.	Rdm.	Ufm.	Rdm.	Ufm.	Rdm.
MQ + Carlson	49	47	9.93E+03	1.20E+04	43	46	1.06E+04	8.56E+03
PS	1239	996	2.92E+04	2.39E+04	1239	996	2.92E+04	2.39E+04
TPS	295	243	1.40E+04	1.32E+04	295	243	1.40E+04	1.32E+04
WU1	41	39	8.26E+03	6.62E+03	41	39	8.26E+03	6.62E+03
WU2	47	47	6.99E+03	9.16E+03	47	47	6.99E+03	9.16E+03
WU3	87	83	9.01E+03	9.35E+03	87	83	9.01E+03	9.35E+03
WU4	689	662	9.29E+03	8.32E+03	689	662	9.29E+03	8.32E+03
WL1	245	234	9.62E+03	9.77E+03	245	234	9.62E+03	9.77E+03
WL2	64	62	9.28E+03	8.53E+03	64	62	9.28E+03	8.53E+03
WL3	51	50	7.02E+03	7.86E+03	51	50	7.02E+03	7.86E+03
WL4	511	504	9.25E+03	7.95E+03	511	504	9.25E+03	7.95E+03
WL5	100	94	1.23E+04	1.01E+04	100	94	1.23E+04	1.01E+04
WL6	69	65	1.07E+04	8.81E+03	69	65	1.07E+04	8.81E+03
WL7	64	62	8.71E+03	8.57E+03	64	62	8.71E+03	8.57E+03
BUH1	117	113	1.13E+04	1.01E+04	117	113	1.13E+04	1.01E+04
BUH2	83	78	1.22E+04	9.22E+03	83	78	1.22E+04	9.22E+03
BUH3	164	154	1.10E+04	1.24E+04	164	154	1.10E+04	1.24E+04

From all the numerical results obtained so far and in addition to the criteria stated, it has to be acknowledged that 'underfitting and overfitting' is another crucial figure to be considered. Regarding all cases, a significant reduction in accuracy, measured by both error norms, produced by PS type of RBF strongly indicates its overfitting nature and shall not be recommended for practical uses. On the other hand, WL5, BUH1, WU3, and BUH2 are seen to be slightly underfitting for both numerical demonstrations. The best types of RBF in terms of this aspect are WU1, WU2, WL1, WL4, and MQ.

5 Conclusions

This work aims to provide insights into the use of sixteen forms of radial basis function (RBF) containing no shape parameter, so they are referred to as 'shapeless RBFs'. The challenge under the main experiment is the problem of pattern recognition through neural networks computation process and architecture. The ability to deal with large and noised datasets of each shapeless RBF is measured under several criteria against the well-known shape-containing RBF called multiquadric (MQ). Two testing functions are tackled numerically and important findings observed (based on each criterion) are listed below.

1) Accuracy (L_∞ and L_{RMSE}): The best three shapeless RBFs are WL6, WL7, and BUH3, whereas PS, WU1, and MQ are found to produce the least accurate results.

2) The condition number ($Cond_\infty(-)$): The best three are WU1, WU2, and WL3 with the comparatively worse ones being PS, TPS, and WL5.

3) In terms of CPU (time and storage) and also the number of basis functions (m): MQ, WL3, and WL6 are seen to be the most desirable whereas PS, WU4, and WL4 are not so promising under these criteria.

4) User's Interference: It is obvious that as long as no parameter-turning process is required, there is no need to interfere with the algorithm and this is one desirable aspect of using shapeless RBF. This is also the case for the sensitivity-to-parameter criteria.

5) Ease of implementation: It is observed that all shapeless RBF types under this work are equally simple when it comes to implementing the scheme. For MQ, on the other hand, an additional coding routine may be needed for the process of a reliable shape searching procedure, as is the case in this investigation for the Carlson algorithm.

Together with the 'overfitting and underfitting' aspect, this work suggests that shapeless RBFs from Wu's and Wendland's families are highly promising, whereas the rest forms are still skeptical for practical uses. Apart from this useful piece of information for pattern recognition application, it might be considered a weakness of the work that the figure discovered so far may change when dealing with other kinds of applications such as direct interpolation, function approximation, and recovery, as well as solving different equations (DEs) under the concepts of meshless or meshfree methods. Furthermore, other kinds of node distribution of sizes may result in different aspects. This is all set as our future research direction and is highly recommended for interested researchers to further explore.

Funding Statement: The authors received funding for this study from the Institute for the Promotion of Teaching Science and Technology, under the Development and Promotion of Science and Technology Talent Project (DPST), Thailand.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Parasher, S. Sharma, A. K. Sharma and J. P. Gupta, "Anatomy on pattern recognition," *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 2, no. 3, pp. 371–378, 2011.

[2] M. Paolanti and E. Frontoni, "Multidisciplinary pattern recognition applications: A review," *Computer Science Review*, vol. 37, pp. 1–23, 2020.

[3] S. Fouladi, A. A. Safaei, N. Mammone, F. Ghaderi and M. J. Ebadi, "Efficient deep neural networks for classification of Alzheimer's disease and mild cognitive impairment from scalp EEG recordings," *Cognitive Computation*, vol. 14, pp. 1247–1268, 2022.
[4] S. Fouladi, M. J. Ebadi, A. A. Safaei, M. Y. Bajuri and A. Ahmadian, "Efficient deep neural networks for classification of COVID-19 based on CT images: Virtualization via software defined radio," *Computer Communications*, vol. 176, pp. 234–248, 2021.
[5] M. J. Ebadi, A. Hosseini and M. M. Hosseini, "A projection type steepest descent neural network for solving a class of nonsmooth optimization problems," *Neurocomputing*, vol. 235, pp. 164–181, 2017.
[6] M. J. D. Powell, "Radial basis function approximations to polynomials," in *Numerical Analysis 87*, NY, United States: Longman Publishing Group, pp. 223–241, 1989.
[7] M. J. D. Powell, "Radial basis functions for multivariable interpolation: A review," in *Algorithms for Approximation*, Oxford, United States: Clarendon Press, pp. 143–167, 1987.
[8] J. Chen, Q. Li, H. Wang and M. Deng, "A machine learning ensemble approach based on random forest and radial basis function neural network for risk evaluation of regional flood disaster: A case study of the Yangtze river delta, China," *International Journal of Environmental Research and Public Health*, vol. 17, no. 1, pp. 1–21, 2019.
[9] R. Wang, D. Li and K. Miao, "Optimized radial basis function neural network based intelligent control algorithm of unmanned surface vehicles," *Journal of Marine Science and Engineering*, vol. 8, no. 3, pp. 1–13, 2020.
[10] C. W. Dawson, "Sensitivity analysis of radial basis function networks for river stage forecasting," *Journal of Software Engineering and Applications*, vol. 13, no. 12, pp. 327–347, 2020.
[11] N. Hemageetha and G. M. Nasir, "Radial basis function model for vegetable price prediction," in *Proc. of the 2013 Int. Conf. on Pattern Recognition, Informatics and Mobile Engineering*, Salem, India, pp. 424–428, 2013.
[12] K. A. Shastry, H. A. Sanjay and G. Deexith, "Quadratic-radial-basis-function-kernel for classifying multi-class agricultural datasets with continuous attributes," *Applied Soft Computing*, vol. 58, pp. 65–74, 2017.
[13] K. A. Rashedi, M. T. Ismail, N. N. Hamadneh, S. AL Wadi, J. J. Jaber *et al.,* "Application of radial basis function neural nnetwork coupling particle swarm optimization algorithm to classification of Saudi Arabia stock returns," *Journal of Mathematics*, vol. 2021, no. 5, pp. 1–8, 2021.
[14] C. Fragopoulos, A. Pouliakis, C. Meristoudis, E. Mastorakis, N. Margari *et al.,* "Radial basis function artificial neural network for the investigation of thyroid cytological lesions," *Journal of Tyroid Research*, vol. 2020, no. 1, pp. 1–14, 2020.
[15] A. Krowiak and J. Podgórski, "On choosing a value of shape parameter in radial basis function collocation methods," in *Int. Conf. of Numerical Analysis and Applied Mathematics (ICNAAM 2018), AIP Conf. Proc.*, Rhodes, Greece, vol. 2116, no. 1, pp. 450020–1–450020–4, 2019.
[16] S. Zheng, R. Feng and A. Huang, "The optimal shape parameter for the least squares approximation based on the radial basis function," *Mathematics*, vol. 8, no. 11, pp. 1–20, 2020.
[17] S. Kaennakham, P. Paewpolsong, N. Sriapai and S. Tavaen, "Generalized-multiquadric radial basis function neural nnetworks (RBFNs) with variable shape parameters for function recovery," *Frontiers in Artificial Intelligence and Applications*, vol. 340, pp. 77–85, 2021.
[18] R. Cavoretto, A. D. Rossi, M. S. Mukhametzhanov and Y. D. Sergeyev, "On the search of the shape parameter in radial basis functions using univariate global optimization methods," *Journal of Global Optimization*, vol. 79, no. 2, pp. 305–327, 2021.
[19] S. Tavaen, K. Chanthawara and S. Kaennakham, "A numerical study of a compactly-supported radial basis function applied with a collocation meshfree scheme for solving PDEs," *Journal of Physics: Conference Series*, vol. 1489, pp. 1–13, 2020.
[20] S. Tavaen and S. Kaennakham, "A comparison study on shape parameter selection in pattern recognition by radial basis function neural networks," *Journal of Physics: Conference Series*, vol. 1921, pp. 1–10, 2021.
[21] S. Tavaen, R. Viriyapong and S. Kaennakham, "Performances of non-parameterised radial basis functions in pattern recognition applications," *Journal of Physics: Conference Series*, vol. 1706, pp. 1–9, 2020.

[22] R. E. Carlson and T. A. Foley, "The parameter R2 in multiquadric interpolation," *Computers & Mathematics with Applications*, vol. 21, no. 9, pp. 29–42, 1991.
[23] Z. Wu, "Compactly supported positive definite radial basis functions," *Advances in Computational Mathematics*, vol. 4, pp. 283–292, 1995.
[24] H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Advances in Computational Mathematics*, vol. 4, pp. 389–396, 1995.
[25] M. D. Buhmann, "Radial functions on compact support," *Proceedings of the Edinburgh Mathematical Society*, vol. 41, no. 1, pp. 33–46, 1998.
[26] M. Shin and C. Park, "A radial basis function approach to pattern recognition and its applications," *ETRI Journal*, vol. 22, no. 2, pp. 1–10, 2000.
[27] H. Wendland, "Stability," in *Scattered Data Approximation*, Cambridge, United Kingdom: Cambridge University Press, pp. 206–222, 2004.
[28] D. Lazzaro and L. B. Montefusco, "Radial basis functions for the multivariate interpolation of large scattered data sets," *Journal of Computational and Applied Mathematics*, vol. 140, no. 1–2, pp. 521–536, 2002.
[29] R. Franke, "Scattered data interpolation: Tests of some method," *Mathematics of Computation*, vol. 38, no. 157, pp. 181–200, 1982.
[30] R. J. Renka and R. Brown, "Algorithm 792: Accuracy tests of ACM algorithms for interpolation of scattered data in the plane," *ACM Transactions on Mathematical Software*, vol. 25, no. 1, pp. 78–94, 1999.