Kybernetika

Jan Kalina; Petra Vidnerová; Patrik Janáček

Highly robust training of regularizedradial basis function networks

*Kybernetika*, Vol. 60 (2024), No. 1, 38–59

Persistent URL: http://dml.cz/dmlcz/152345

KYBERNETIKA — VOLUME *60* (2024), NUMBER 1, PAGES 38–59

# HIGHLY ROBUST TRAINING OF REGULARIZED RADIAL BASIS FUNCTION NETWORKS

Jan Kalina, Petra Vidnerová and Patrik Janáček

Radial basis function (RBF) networks represent established tools for nonlinear regression modeling with numerous applications in various fields. Because their standard training is vulnerable with respect to the presence of outliers in the data, several robust methods for RBF network training have been proposed recently. This paper is interested in robust regularized RBF networks. A robust inter-quantile version of RBF networks based on trimmed least squares is proposed here. Then, a systematic comparison of robust regularized RBF networks follows, which is evaluated over a set of 405 networks trained using various combinations of robustness and regularization types. The experiments proceed with a particular focus on the effect of variable selection, which is performed by means of a backward procedure, on the optimal number of RBF units. The regularized inter-quantile RBF networks based on trimmed least squares turn out to outperform the competing approaches in the experiments if a highly robust prediction error measure is considered.

*Keywords:* regression neural networks, robust training, effective regularization, quantile regression, robustness

*Classification:* 68T37, 68W25, 62J02

## 1. INTRODUCTION

Standard tools for training common types of shallow neural networks are vulnerable with respect to the presence of outliers (atypical/anomalous instances) in the data. This is true for multilayer perceptrons (MLPs) and at the same time for radial basis function (RBF) networks, where both represent established classes of feedforward networks for nonlinear regression modeling with numerous applications [1].

There have been remarkably more robust tools available for training MLPs compared to RBF networks [10]. For MLPs, approaches based on robust loss functions inspired by robust or nonparametric statistical procedures [15] have been revealed in [24] to outperform competing robust approaches, e. g. those based on a prior outlier detection and removal. Particularly, robust approaches based on least trimmed squares or least trimmed absolute values estimators turn out to yield superior results for data contaminated by outliers [35]. These highly robust approaches have a high breakdown point in both linear and nonlinear regression [17]; the breakdown point as a fundamental measure

DOI: 10.14736/kyb-2024-1-0038

of robustness is formally defined as the minimal fraction of data that can drive an estimator beyond all bounds when set to arbitrary values (Section 2.6 of [15]). Regularized versions of highly robust MLPs were systematically compared in [18], where however the potential of MLPs to estimate nonlinear regression quantiles was not exploited.

The RBF networks can be described as distribution mixtures and important examples of generalized regularization networks; these were investigated already in [31] and other their examples include tensor product splines or adaptive splines. Regression RBF networks also require robust training and an important class of such available robust approaches is based on replacing the most common loss function (sum of squared residuals) by a more robust version. Reinforced learning for an RBF network with a robustified loss function using softmax-based iterative quadratic programming was used in [47]. A generalized Kullback-Leibler divergence was used as the loss function in training RBF networks in [38]; this turned out to be more suitable than a habitual approach if the random noise is not Gaussian.

Practitioners seem to realize the harmful influence of outliers on the standard training of RBF networks. Let us recall two recent applications, where a standard training of an RBF network was accompanied by outlier detection. A prior detection of outliers and their removal (before the training) were performed in [43] in the task of modeling wireless sensor network data. A posterior outlier detection was performed in [11] as a diagnostic tool to validate results of RBF networks predicting the quality of crude oil samples from various geographic locations.

Let us also recall some other approaches for improving the robustness of regression RBF networks, neither based on a robust loss function nor on outlier detection. Subtractive clustering was used in [40] for the search of center vectors for RBF networks and this was claimed to partially improve the robustness against outliers, because it is focused on approximating the underlying mapping rather than on interpolating the training data. Attempts to robustify RBF networks based on exploiting the Mahalanobis distances of individual data points from the centers were described in [2]; however, the Mahalanobis distances remain non-robust with respect to the breakdown point [3]. In the credit rating application of [26], an optimized segmentation of customers was performed aimed at reducing the vulnerability of an RBF network to outliers. Some other works combined several approaches to improving the robustness of RBF networks to outliers. For example, a composite loss function in the form of a sequence of sigmoidal functions was used in [25] to improve the robustness together with using the redescending M-estimator of Hampel (Section 3.2 of [15]) for the parameter estimation. It deserves to be recalled that a number of applied papers (e. g. [19]) endeavored for robust performance of RBF networks, i. e. stable performance under modified conditions, however without using any statistical robustness focused on the effect of outliers.

The predictive ability of robust RBF networks has been compared only on a very small number of datasets with outliers [47] so that robust RBF networks started to penetrate to real applications only slowly. Moreover, the literature is void of regularized versions of robust RBF networks, although various regularization types have been often exploited for the plain (non-robust) RBF networks [27]. This paper is interested in highly robust regularized versions of RBF networks. Section 2 is methodological and contains several novel proposals including a robust inter-quantile version of RBF networks based on the

trimmed least squares. An analysis performed over 405 trained networks is presented in Section 3. A more detailed analysis of the effect of variable selection is presented on two real datasets in Section 4. Based on the computations, the novel inter-quantile RBF network based on the trimmed least squares may be recommended for real data, especially if combined with the $L_2$-regularization. Section 5 brings conclusions.

## 2. METHODS

The standard nonlinear regression model is considered here in the form

$$Y_i = f(X_i) + e_i, \quad i = 1, \ldots, n, \tag{1}$$

with the total number of $n$ observations (data samples), where $Y_1, \ldots, Y_n$ are values of the response and $X_1, \ldots, X_n \in \mathbb{R}^p$ are the regressors (predictors). The random errors $e_1, \ldots, e_n$ are assumed to independent and identically distributed (i.i.d.) but we do not want to assume any particular probabilistic distribution of the errors. The regression task is to explain (model, predict) the unknown nonlinear function $f$ based on the data.

### 2.1. Radial basis function networks

The plain RBF network for the task (1) requires to choose a radially symmetric function $\rho$. The model, which was described in detail e. g. in [1], has the form

$$f(x) = \sum_{j=1}^{N} a_j \rho(\|x - c_j\|) \tag{2}$$

for a given value of the features $x \in \mathbb{R}^p$, where $\|.\|$ denotes the Euclidean norm, $c_1, \ldots, c_N \in \mathbb{R}^p$ are centers (center vectors), and $a_1, \ldots, a_N \in \mathbb{R}$ are parameters denoted as weights. The network (2) represents a weighted sum of distances of $x \in \mathbb{R}^p$ from a (possibly small) set of important points in the $p$-dimensional space. The role of the centers in RBF networks was discussed in [8], where different approaches to their optimization were carefully compared. It is recommended to start the optimization with centers corresponding to particular (randomly selected) observations. A denser set of centers is typically required in the regions in the space of the data with a highly nonlinear and strongly curved course of the regression function.

We consider here the most common approach with the quadratic loss

$$\arg\min \frac{1}{n} \sum_{i=1}^{n} \left( Y_i - \sum_{j=1}^{N} \hat{a}_j \rho(\|X_i - \hat{c}_j\|) \right)^2, \tag{3}$$

which is minimized jointly over $\hat{c}_1, \ldots, \hat{c}_N \in \mathbb{R}^p$, $\hat{a}_1, \ldots, \hat{a}_N \in \mathbb{R}$, and over other possible parameters corresponding to $\rho$. The parameters $c_1, \ldots, c_N$ and $a_1, \ldots, a_N$ are estimated here by estimates $\hat{c}_1, \ldots, \hat{c}_N$ and $\hat{a}_1, \ldots, \hat{a}_N$, respectively. The number $N$ of RBF units (typically assuming $p < N < n$) is fixed in (3) and can be tuned in cross-validation.

It will be convenient to introduce a schematic notation for (3). Let us denote the vector of all parameters of a given RBF network by $\theta \in \mathbb{R}^d$ with a certain $d$. If estimating $\theta$ by $\hat{\theta} \in \mathbb{R}^d$, the residuals will be denoted as $u_i(\hat{\theta})$ for $i = 1, \ldots, n$. Then, we express (3) as

$$\arg\min_{\hat{\theta} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} u_i^2(\hat{\theta}). \tag{4}$$

To avoid confusion, the plain RBF network will be denoted as LS-RBF throughout the paper, where LS abbreviates the least squares, i. e. reflects the loss to consider the sum of squared residuals. Among various available approaches to estimating the optimal values of the parameters, we have chosen back-propagation exploiting a stochastic gradient descent optimization algorithm. This can be used for all of the robust versions of RBF networks mentioned below. Because the computation requires to specify starting points for the parameters, we use the (typical) approach with weights initialized to random numbers close to 0.

Further, we consider the habitual approach to take $\rho$ to be the Gaussian kernel (Gaussian density). With such choice, (2) can be expressed as

$$f(x) = \sum_{j=1}^{N} a_j \exp\left\{ -\frac{\|x - c_j\|^2}{2\sigma_j^2} \right\}, \quad x \in \mathbb{R}^p, \tag{5}$$

and the training requires to estimate also the unknown quantities $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_N^2$, i. e. variability parameters or bandwidth of individual Gaussian kernels.

### 2.2. RBF networks with a robust loss

A simple robustified version of RBF networks based on computing

$$\arg\min_{\hat{\theta} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} |u_i(\hat{\theta})| \tag{6}$$

will be denoted as LAD-RBF, where LAD abbreviates the least absolute deviations (also known as the least absolute values estimator) [46]. Other available RBF networks with a robust loss, which were recently proposed in the literature, have been inspired by robust statistical estimators from the linear regression model. Such tools include the least weighted squares (LWS) estimator [44] or its important special case, which is the least trimmed squares (LTS) [50]. In linear regression, rank-based estimators performed successfully in applications [16, 17, 18] or in simulated measurement error models [37], while the results were not heavily depending on the choice of the particular weight function.

Let us use the notation

$$u_{(1)}^2(\hat{\theta}) \leq \cdots \leq u_{(n)}^2(\hat{\theta}) \tag{7}$$

for squared residuals arranged in asceding order. A robust RBF network based on the LWS [17] denoted as LWS-RBF is defined by

$$\arg\min_{\hat{\theta} \in \mathbb{R}^d} \sum_{i=1}^{n} \psi\left( \frac{i - 1/2}{n} \right) u_{(i)}^2(\hat{\theta}), \tag{8}$$

where the weight function is considered as in [18] in the form

$$\psi_1(t) = \exp\left\{-\frac{t^2}{2\tau^2}\right\} \mathbb{1}[t < \alpha], \quad t \in [0,1], \quad \tau = 0.8, \quad \alpha = 0.75. \tag{9}$$

The function (9) can be interpreted as a trimmed density of the standard Gaussian distribution (up to a multiplicative constant). As a special case, LTS-RBF is defined as (8) using

$$\psi_2(t) = \mathbb{1}[t < \alpha], \quad t \in [0,1], \quad \alpha = 0.75, \tag{10}$$

where $\mathbb{1}$ denotes the indicator function.

As a novelty, let us propose LWA-RBF networks defined by

$$\underset{\hat{\theta}\in\mathbb{R}^d}{\arg\min} \sum_{i=1}^{n} \psi\left(\frac{i-1/2}{n}\right) |u(\hat{\theta})|_{(i)} \tag{11}$$

again with the weight function (9), where

$$|u(\hat{\theta})|_{(1)} \leq \cdots |u(\hat{\theta})|_{(n)}. \tag{12}$$

The least trimmed absolute value (LTA) estimator [35] is an important special case of LWA; we propose a corresponding robust RBF network denoted as LTA-RBF as (11) with the particular weight function (10).

### 2.3. Regularized versions of robust RBF networks

Regularization techniques have been recommended for non-robust RBF networks to avoid overfitting [13] for data with a smaller value of $n$, particularly if $p$ is not too small. The most common regularization approach exploits the weight regularization (weight decay) in the form $L_2$- or $L_1$-regularization considered only for the values $a_1, \ldots, a_N$ in (5). However, the combination of robustness with regularization for RBF networks has not been considered in the literature. Let us denote the selected loss function of an RBF network by $\ell(\hat{\theta})$; we understand that $\hat{\theta}$ contains (apart from other estimates) also the estimates $\hat{a}_1, \ldots, \hat{a}_N$ and $\hat{c}_1, \ldots, \hat{c}_N$. Let us consider the $L_2$-regularization in the form

$$\underset{\hat{\theta}\in\mathbb{R}^d}{\arg\min} \left[\ell(\hat{\theta}) + \lambda \sum_{j=1}^{N} \hat{a}_j^2\right], \tag{13}$$

where a suitable value of the regularization parameter $\lambda > 0$ may be found by cross-validation. $L_1$-regularization, which induces sparsity of the parameters, uses the penalization term $\lambda \sum_{j=1}^{N} |\hat{a}_j|$.

Regularization in the form of penalization allows RBF networks to be computed also for high-dimensional data with $n < p$ and may also be used for other robust versions of RBF networks. Because the regularization improves local robustness of the standard LS-RBF [45] but not robustness to outliers, we expect robust regularized versions of RBF networks such as (13) to be the appropriate tools for contaminated data.

### 2.4. RBF networks based on quantiles (TLS-RBF network)

In linear regression, regression quantiles represent a popular methodology for obtaining more complex information compared to fitting a single regression hyperplane [23, 12]. The trimmed least squares (TLS) estimator in linear regression was proposed already in [32] as one of robust regression $L$-estimators. An adaptive (data-dependent) but symmetric version, which is based on trimming the same percentage of outliers from above and from below, was studied intensively in Chapters 4 and 6 of [7]. Later, the asymptotic representation for the TLS estimator, which does not require the both percentages to be equal, was proven in [15] together with its appealing robustness properties. We are interested in extending the adaptive version of the TLS estimator to the context of neural networks; a robust trimmed least squares RBF network (TLS-RBF) based on finding the optimal values of its two parameters is proposed in this section.

The quantile regression RBF networks denoted here shortly as QR-RBF, which were suggested e. g. in the papers [22, 49], represent a method for estimating quantiles of nonlinear trend inspired by [30]. Here, QR-RBF($\tau$) denotes the quantile with $\tau \in (0, 1)$ defined by

$$\arg\min_{\hat{\theta}\in\mathbb{R}^d} \sum_{i=1}^{n} \zeta_\tau \left( Y_i - \sum_{j=1}^{N} \hat{a}_j \psi(\|X_i - \hat{c}_j\|) \right), \tag{14}$$

i. e. an RBF network with the loss function $\zeta_\tau$ with a given $\tau$. Compared to the minimization in (3), the so-called check function $\zeta_\tau$ is used with the definition

$$\zeta_\tau(x) = \begin{cases} (\tau - 1)x, & \text{if } x < 0, \\ \tau x, & \text{if } x \geq 0. \end{cases} \tag{15}$$

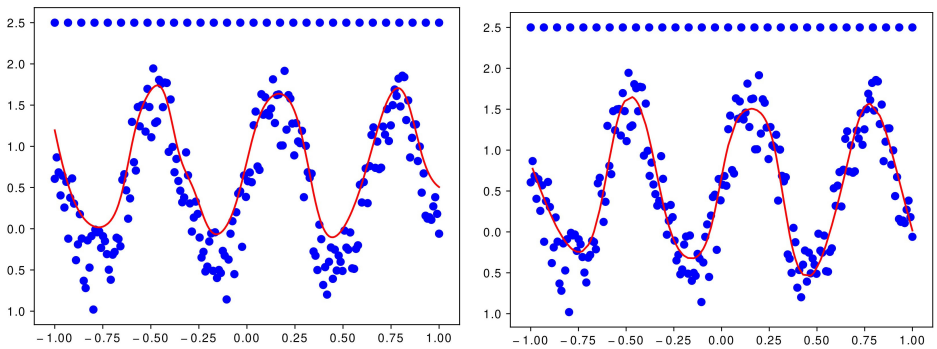


**Fig. 1.** Artificial data with trend estimated by LS-RBF (left) and TLS-RBF with $\tau = 0.3$ and $\tau = 0.7$ (right).

The idea of the novel method TLS-RBF defined in Algorithm 1 is to train an LS-RBF network over only the training observations that lie between two QR-RBF networks with parameters $\tau_1$ and $\tau_2$. The method performs outlier detection and the parameters $\tau_1 < \tau_2$

are tuned according to the estimated contamination level. Detecting and trimming away the outlying training observations ensures the robustness. TLS-RBF represents the first available nonlinear extension of TLS and may be useful in applications, where the primary aim is not related to estimating individual nonlinear quantiles, especially if the errors in (1) are heteroscedastic [34].

**Algorithm 1** TLS-RBF network

**Input:** Training data $(X_1^T, Y_1)^T, \ldots, (X_n^T, Y_n)^T$, where $X_i \in \mathbb{R}^p$ and $Y_i \in \mathbb{R}$ for $i = 1, \ldots, n$

**Input:** Integer $k$ (with default value $k = 10$)

**Output:** TLS-RBF with values of parameters $\tau_1$ and $\tau_2$ that are optimal in $k$-fold cross-validation

**for** $j \in 1, \ldots, k$ **do**
  Divide the data randomly to $k$ groups of approximately equal size
  $S_j :=$ set of observations of the $j$th group
  **for** $\tau_1 \in \{0.05, 0.10, \ldots, 0.25\}$ **do**
    **for** $\tau_2 \in \{0.95, 0.90, \ldots, 0.75\}$ **do**
      Train the QR-RBF($\tau_1$) and QR-RBF($\tau_2$) networks over all data $\notin S_j$
      $\hat{Y}_i^{\mathsf{QR\text{-}RBF}(\tau_1)} :=$ fitted value of $Y_i$ given by QR-RBF($\tau_1$) for data $\notin S_j$
      $\hat{Y}_i^{\mathsf{QR\text{-}RBF}(\tau_2)} :=$ fitted value of $Y_i$ given by QR-RBF($\tau_2$) for data $\notin S_j$
      TLS-RBF($\tau_1, \tau_2$):= LS-RBF computed over the observations that

$$\hat{Y}_i^{\mathsf{QR\text{-}RBF}(\tau_1)} \leq Y_i \leq \hat{Y}_i^{\mathsf{QR\text{-}RBF}(\tau_2)}, \tag{16}$$

  where $Y_i \notin S_j, \quad i = 1, \ldots, n$
      MSE($j, \tau_1, \tau_2$) := MSE of TLS-RBF($\tau_1, \tau_2$) for data $\in S_j$
    **end for**
  **end for**
**end for**
MSE($\tau_1, \tau_2$) := $\sum_{j=1}^{10}$ MSE($j, \tau_1, \tau_2$)

$$(\hat{\tau}_1, \hat{\tau}_2)^T := \arg\min_{\tau_1, \tau_2} \mathsf{MSE}(\tau_1, \tau_2) \tag{17}$$

Train the QR-RBF($\hat{\tau}_1$) and QR-RBF($\hat{\tau}_2$) networks over all data
$\hat{Y}_i^{\mathsf{QR\text{-}RBF}(\hat{\tau}_1)} :=$ fitted value of $Y_i$ given by QR-RBF($\hat{\tau}_1$) for all data
$\hat{Y}_i^{\mathsf{QR\text{-}RBF}(\hat{\tau}_2)} :=$ fitted value of $Y_i$ given by QR-RBF($\hat{\tau}_2$) for all data
TLS-RBF($\hat{\tau}_1, \hat{\tau}_2$):= LS-RBF network computed over the observations that

$$\hat{Y}_i^{\mathsf{QR\text{-}RBF}(\hat{\tau}_1)} \leq Y_i \leq \hat{Y}_i^{\mathsf{QR\text{-}RBF}(\hat{\tau}_2)}, \quad i = 1, \ldots, n \tag{18}$$

### 2.5. Backward variable selection

Backward elimination (backward variable selection) represents a popular step-wise variable selection approach in the context of linear regression [4], which has been successfully exploited also for RBF networks (recently e. g. in [14]). A standard version is used here to find relevant variables based on the value of cross-validation prediction error measure. The variable selection is used only in Section 4 in a detailed study over two selected datasets; it is used as a prior step before the training of RBF networks. The variable selection is performed by selecting and trimming away redundant regressors from the model. In the algorithm, the notation $\mathrm{RBF}_{\ell,N,\lambda}$ is used for a version of an RBF network with loss function $\ell$, $N$ RBF units, and regularization parameter $\lambda$. If the network is computed over all variables except for those contained in a set $S$, it will be convenient to denote it by $\mathrm{RBF}_{\ell,N,\lambda}(S)$.

Within the backward procedure, values of $N$ and $\lambda$ are not optimized repeatedly. Instead, the tedious computations are simplified by running the whole backward selection with a single choice of $N$ and $\lambda$ optimized for each individual dataset. At the end of the backward selection, the number of regressors $p$ is reduced to a smaller number denoted as $\tilde{p}$; the optimal $N$ is determined again after the variable selection.

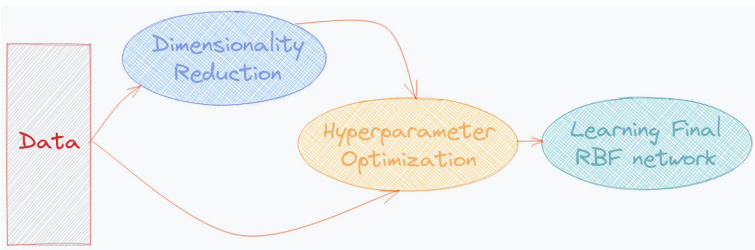


**Fig. 2.** A general scheme for training RBF networks. Dimensionality reduction is used only in the experiments of Section 4.

## 3. EXPERIMENTS OVER 405 TRAINED RBF NETWORKS

The first study is performed without any dimensionality reduction and uses 15 publicly available datasets with $n$ between 151 and 9358 and with $p$ between 3 and 128. Nonlinear regression represents a meaningful task for these datasets, which have been previously used as benchmarking datasets for regression modeling. All the computations in this paper were performed in Keras [6].

### 3.1. Experiments

For each dataset, the response is always normalized to the interval $[0, 1]$; this allows to aggregate the results obtained for different datasets. Only continuous or binary regressors are retained ignoring the other ones. Further, all observations with at least one missing value were omitted, i. e. we perform a so-called complete case analysis. After

| Index | Name of the dataset | $p$ | $n$ | Source |
|---|---|---|---|---|
| 1 | Air Quality | 15 | 9358 | [9] |
| 2 | Auto MPG | 8 | 398 | [9] |
| 3 | Bennett5 | 3 | 154 | [28] |
| 4 | Boston Housing | 14 | 506 | [9] |
| 5 | Communities and Crime | 128 | 1994 | [9] |
| 6 | Concrete Compressive Strength | 9 | 1030 | [9] |
| 7 | Chwirut1 | 3 | 214 | [28] |
| 8 | ENSO | 9 | 168 | [28] |
| 9 | Gauss1 | 8 | 250 | [28] |
| 10 | Hahn1 | 7 | 236 | [28] |
| 11 | Kirby2 | 5 | 151 | [28] |
| 12 | Parkinsons Telemonitoring | 26 | 5875 | [9] |
| 13 | SkillCraft1 Master Table | 20 | 3395 | [9] |
| 14 | SML2010 | 24 | 4137 | [9] |
| 15 | Solar Flare | 10 | 1389 | [9] |

**Tab. 1.** The datasets considered in the experiments of Section 3.

these steps are performed, each dataset is divided to a training set and a test set, where the test set is randomly chosen to contain 20 % of the samples.

In the training data, 10-fold cross-validation is performed to find the optimal values of $N$ and $\lambda$ (only for regularized methods); the pair of $N$ and $\lambda$ that yields the minimal cross-validation MSE is selected, where $N$ is considered from the set $\{100, 200, 400\}$ and $\lambda$ is from the set $\{0.0001, 0.001, 0.01, 0.1, 0.25\}$. Within the cross-validation, each of the variables was standardized to values between 0 and 1. There are 10 folds for each of the 15 datasets and each type of the network is thus evaluated in $15 \times 10 \times 3 \times 5 = 2250$ situations, i. e. for 3 values of $N$ and 5 values of $\lambda$.

After the optimal $N$ and $\lambda$ are found, the trained RBF networks are applied to the test sets to perform independent validation of the performance of different methods: there were 9 methods (9 types of networks given in Table 3) and 3 types of regularization applied to train the network and to make predictions for the test sets of 15 datasets. At this point, there were thus altogether $9 \times 3 \times 15 = 405$ trained versions of networks used in the analysis. While $\lambda$ is optimized for MSE, we compute MAE, TMSE, and TMAE with the value of $\lambda$ that is optimal for MSE.

MLPs and LWS-MLPs [18] are also trained for the sake of comparisons. They are used with 2 hidden layers containing $T$ and 2 neurons, respectively, with a sigmoid activation function in the hidden layers, and with a linear output layer with a single neuron. As values of $T$, all numbers from the set $S_T = \{25, 50, 100, 200, 400\}$ have been used; the results are then considered for the value of $T$ that yields the smallest loss function among all possible values from $S_T$. An $L_2$-regularized version of MLPs is considered as well, because $L_2$-regularization turned out to be the best choice among various regularization techniques in [18].

| | LS-RBF | LAD-RBF | TLS-RBF | LTS-RBF | LTA-RBF | LWS-RBF | LWA-RBF | MLP | LWS-MLP |
|---|---|---|---|---|---|---|---|---|---|
| LS-RBF | | | | | | | | | |
| LAD-RBF | 70 | | | | | | | | |
| TLS-RBF | 89 | 77 | | | | | | | |
| LTS-RBF | 87 | 74 | 36 | | | | | | |
| LTA-RBF | 87 | 72 | 32 | 46 | | | | | |
| LWS-RBF | 87 | 73 | 35 | 58 | 53 | | | | |
| LWA-RBF | 85 | 73 | 30 | 43 | 48 | 39 | | | |
| MLP | 44 | 24 | 11 | 16 | 19 | 21 | 19 | | |
| LWS-MLP | 88 | 78 | 35 | 55 | 46 | 41 | 44 | 87 | |

**Tab. 2.** Results of pairwise comparisons of Section 3 for TMSE aggregated over all different combinations of regularization types, values of $\lambda$, and datasets. Each value is a percentage of experiments, where the row method is better (has a smaller TMSE) than the column method.

### 3.2. Prediction error measures

Four prediction error measures are used to evaluate the prediction performance of the trained networks. Their values computed for each of the test sets are averaged across the 15 datasets. Using robust versions of the mean square error (MSE) is motivated by the vulnerability of MSE with respect to outliers. The definitions of the used measures will be now recalled using the notation $m$ for the number of samples in a particular test set and $u_1, \ldots, u_m$ for the prediction errors (residuals) of the test samples.

- Mean square error $\text{MSE} = \sum_{i=1}^{m} u_i^2/m$.
- Mean absolute deviation $\text{MAE} = \sum_{i=1}^{n} |u_i|/m$.
- Trimmed mean square error (TMSE) and trimmed mean absolute error (TMAE). We use the notation (7) and denote $|u|_{(1)} \le \cdots \le |u|_{(m)}$ for ordered absolute values. Taking $h = \lfloor 3m/4 \rfloor$, where $\lfloor x \rfloor$ denotes the integer part of $x \in \mathbb{R}$, we define

$$\text{TMSE} = \frac{1}{h}\sum_{i=1}^{h} u_{(i)}^2 \quad \text{and} \quad \text{TMAE} = \frac{1}{h}\sum_{i=1}^{h} |u|_{(i)}. \tag{19}$$

### 3.3. Results

As revealed in Table 3, robust RBF networks turn out to beneficial and also the regularization is beneficial for them; the choice of the best regularization type depends on the choice of the particular loss function. We do not consider MSE and MAE to be appropriate for data influenced by severe outliers and TLS-RBF turns out to outperform other networks if using highly robust prediction error measures (TMSE, TMAE). For

both TMSE and TMAE, the best results are obtained for TLS-RBF if $L_2$-regularization is used. The fact that $L_2$-regularization is so beneficial corresponds to the experience described in the literature on regularized RBF networks [33]. Results of LWS-RBF, LWA-RBF, LTS-RBF, and LTA-RBF stay slightly behind the best results. Results obtained with LWS-MLP much outperform those of non-robust MLP here, but do not reach the level of TLS-RBF.

More detailed results obtained for TMSE are reported in Table 2, which contains results of pairwise comparisons aggregated for 3 types of regularization, 5 values of $\lambda \in \{0.0001, 0.001, 0.01, 0.1, 0.25\}$, and 15 datasets. The results are percentages evaluated across $3 \times 5 \times 15 = 225$ situations. Particularly, each value in the table is the number of situations, where the method in the particular row has a better (smaller) TMSE on the test set compared to the method in the particular column. For example, LWS-RBF yields a smaller TMSE than LS-RBF in 87 % of situations. The table reveals the weak performance of LS-RBF compared to all robust versions of RBF networks and confirms TLS-RBF networks to be the winner of the comparions.

### 3.4. Hypothesis testing

In addition, hypothesis testing was further performed to assess the significance of different loss functions and/or different regularization types. The nonparametric Friedman test applied to the prediction error measures (PEMs) evaluated for individual RBF networks rejects the null hypothesis that the results are equal for all combinations of loss functions and regularization types; the $p$-values are smaller than $10^{-5}$ for MSE, $10^{-6}$ for TMSE, and $10^{-3}$ for both MAE and TMAE.

Because of the significant results of the Friedman test, we may proceed with posthoc tests of equal performance in terms of the 4 PEMs for all three pairwise comparisons, i. e. (i) loss functions, (ii) regularization types, and (iii) complete models with all interactions between individual choices of (i) and (ii). The correction to multiple hypotheses testing is performed for all these comparisons with the Tukey honest significance difference (HSD) test [42], which controls the family-wise error rate (FWER). Comparing the effect of loss function, the posthoc tests are used to compare their $9 \times (9-1)/2 = 36$ pairs. Among them, the effect of the loss function is significant in 21 pairs for all the 4 PEMs. There are 6 pairs that do not significantly differ for any of the 4 PEMs; these are the pairs between RBF networks based on LTS, LTA, LWS, and LWA. Comparing the effect of regularization type, the posthoc tests reveal that $L_2$- and $L_1$-regularizations do not significantly differ for any of the 4 PEMs, and each of them is significantly different from results obtained without regularization for all the 4 PEMs. Finally, the posthoc tests are applied to testing between individual pairs of complete models including interaction between a loss function and a regularization type. Among the $(9 \times 3) \times ((9 \times 3) - 1)/2 = 351$ pairs, there are 212 pairs with significant difference for all the 4 PEMs, and only 39 pairs with insignificant difference for all the 4 PEMs.

| Version of network | Prediction error measure MSE | MAE | TMSE | TMAE |
|---|---|---|---|---|
| | (A) No regularization | | | |
| LS-RBF | 0.434 | 0.341 | 0.387 | 0.296 |
| LAD-RBF | 0.477 | 0.313 | 0.333 | 0.260 |
| TLS-RBF | 0.490 | 0.298 | 0.240 | 0.214 |
| LTS-RBF | 0.603 | 0.330 | 0.271 | 0.225 |
| LTA-RBF | 0.616 | 0.322 | 0.276 | 0.217 |
| LWS-RBF | 0.628 | 0.315 | 0.259 | 0.230 |
| LWA-RBF | 0.611 | 0.317 | 0.272 | 0.218 |
| MLP | 0.471 | 0.336 | 0.395 | 0.272 |
| LWS-MLP | 0.582 | 0.329 | 0.294 | 0.231 |
| | (B) $L_2$-regularization | | | |
| LS-RBF | **0.405** | 0.314 | 0.369 | 0.264 |
| LAD-RBF | 0.443 | 0.287 | 0.307 | 0.253 |
| TLS-RBF | 0.469 | 0.296 | **0.221** | **0.206** |
| LTS-RBF | 0.574 | 0.313 | 0.245 | 0.211 |
| LTA-RBF | 0.592 | 0.309 | 0.264 | 0.208 |
| LWS-RBF | 0.599 | 0.296 | 0.260 | 0.210 |
| LWA-RBF | 0.590 | 0.314 | 0.266 | 0.212 |
| MLP | 0.449 | 0.320 | 0.350 | 0.257 |
| LWS-MLP | 0.546 | 0.305 | 0.278 | 0.220 |
| | (C) $L_1$-regularization | | | |
| LS-RBF | 0.415 | 0.308 | 0.361 | 0.269 |
| LAD-RBF | 0.457 | 0.296 | 0.307 | 0.257 |
| TLS-RBF | 0.478 | **0.281** | 0.228 | 0.220 |
| LTS-RBF | 0.588 | 0.315 | 0.249 | 0.213 |
| LTA-RBF | 0.601 | 0.301 | 0.260 | 0.210 |
| LWS-RBF | 0.606 | 0.311 | 0.253 | 0.211 |
| LWA-RBF | 0.599 | 0.305 | 0.256 | 0.315 |
| MLP | 0.453 | 0.315 | 0.374 | 0.263 |
| LWS-MLP | 0.560 | 0.303 | 0.275 | 0.226 |

**Tab. 3.** Results of Section 3: Values of 4 prediction error measures evaluated for various loss functions and regularization types, averaged across the 15 datasets (evaluated over their test sets). The best result in every column is shown in boldface.

## 4. EXPERIMENTS WITH BACKWARD VARIABLE SELECTION

Further, the effect of backward variable selection is investigated. Because the procedure is computationally tedious, we analyze only two real publicly available datasets with a sufficiently large $p$. These datasets have been repeatedly used as benchmarking datasets for nonlinear regression.

1. The first dataset is the Communities and Crime Dataset from the UCI repository [9]. While two versions of this dataset are available, we take the normalized version of this dataset with $n = 1994$ and $p = 128$. We take the total number of crimes (violent and non-violent) per 100 000 inhabitants as the dependent variable.

2. The second dataset is the Residential Building Dataset from the UCI repository with $n = 372$ and $p = 105$. The actual sales price is used here as the response variable.

In both the datasets, all the variables are continuous and there are no missing values. We normalized all the variables to the interval $[0, 1]$. After that, each dataset is divided to a training set and a test set, where the test set was randomly chosen to contain 20 % of the samples. The datasets are analyzed

(i) Without any dimensionality reduction, and

(ii) Using the prior dimensionality reduction of Section 2.5, which is trained over the training data.

Under (i), the cross-validation for finding the optimal $N$ and $\lambda$ (if applicable) over the training data is used in the same way as in Section 3. Naturally, the trained (standard or robust) RBF networks are evaluated over the test data.

Under (ii), the dimensionality reduction is performed for a particular choice of the type of RBF network and particular regularization. The dimensionality reduction contains a cross-validation for finding the optimal $N$ and $\lambda$. Thus, after the dimensionality reduction is computed yielding the relevant set of $\tilde{p}$ variables, the trained RBF network is evaluated directly over the test data.

The choice of $N$ is the only difference here from the approach of Section 3. This is motivated by the need to fulfil the requirement $p < N < n$. For the Communities and Crime Dataset, the optimal value of $N$ is found from the set $\{200, 400, 800\}$ in both (i) and (ii). For the Residential Building Dataset, $N = 150$ is always taken in both (i) and (ii).

In both (i) and (ii), MLPs are used or the sake of comparisons; the same architecture is used as described in Section 3.1.

The results evaluated over the test sets are presented in Tables 4 and 5. There, different regularization types are used (none, $L_2$, $L_1$). For each method, 3 prediction error measures are reported together with the number of variables used in the model ($p$ or $\tilde{p}$) and the optimal $\lambda$. The values of MSE, MAE, and TMSE in both tables 1 should be multiplied by $10^{-2}$. The left parts of the tables are evaluated over all variables and the right parts after performing the backward variable selection of Section 2.5. In each

**Algorithm 2** Backward variable selection for a (possibly robust and/or regularized) RBF network

**Input:** Training data $(X_1^T, Y_1)^T, \ldots, (X_n^T, Y_n)^T$, where $X_i \in \mathbb{R}^p$ and $Y_i \in \mathbb{R}$ for $i = 1, \ldots, n$

**Input:** Selected type of RBF network given by its loss function $\ell$ including a possible regularization with parameter $\lambda > 0$

**Input:** Prediction error measure PEM $\in$ {MSE, MAE, TMSE, TMAE}

**Input:** $\mathcal{N} = \{200, 400, 800\}$ (Communities and Crime Dataset) or $\mathcal{N} = \{150\}$ (Residential Building Dataset)

**Input:** $\Lambda = \{0.0001, 0.001, 0.01, 0.1, 0.25\}$ for regularized types of RBF networks; otherwise $\Lambda := \{\}$

**Input:** $c = 1.1$

**Output:** Set of selected relevant regressors $\tilde{S}$

**Output:** Optimal $N$ and $\lambda$ for data after backward selection

**Output:** Minimal cross-validation PEM for data after backward selection

**for** $\lambda \in \Lambda$ **do**
  **for** $N \in \mathcal{N}$ **do**
    Fit the $\mathrm{RBF}_{\ell,N,\lambda}$ network
    $\mathrm{PEM}_{\ell,N,\lambda}$ := cross-validation PEM of $\mathrm{RBF}_{\ell,N,\lambda}$
  **end for**
**end for**
$\lambda_0 := \arg\min_{\lambda\in\Lambda} \mathrm{PEM}_{\ell,N,\lambda}$
$N_0 := \arg\min_{N\in\mathcal{N}} \mathrm{PEM}_{\ell,N,\lambda_0}$
$S_0 := \{\}$
**repeat**
  **for** $j \notin S$ **do**
    Fit the $\mathrm{RBF}_{\ell,N_0,\lambda_0,}(S \cup \{j\})$ network
    $\mathrm{PEM}_{\ell,N_0,\lambda_0}(S \cup \{j\})$ := cross-validation PEM of $\mathrm{RBF}_{\ell,N_0,\lambda_0}(S \cup \{j\})$
  **end for**
  $\tilde{j} := \arg\min_{j\notin S} \mathrm{PEM}_{\ell,N_0,\lambda_0} S \cup \{j\}$
  $S := S \cup \{\tilde{j}\}$
**until** $\mathrm{PEM}_{\ell,N_0,\lambda_0}(S) \geq c\,\mathrm{PEM}_{\ell,N_0,\lambda_0}$
$\tilde{S} := S \setminus \{\tilde{j}\}$
**for** $\lambda \in \Lambda$ **do**
  **for** $N \in \mathcal{N}$ **do**
    Fit the $\mathrm{RBF}_{\ell,N,\lambda}(\tilde{S})$ network
    $\mathrm{PEM}_{\ell,N,\lambda}(\tilde{S})$ := cross-validation PEM of $\mathrm{RBF}_{\ell,N,\lambda}(\tilde{S})$
  **end for**
**end for**

$$\begin{pmatrix} \tilde{N} \\ \tilde{\lambda} \end{pmatrix} := \underset{N\in\mathcal{N},\ \lambda\in\Lambda}{\arg\min}\ \mathrm{PEM}_{\ell,N,\lambda}(\tilde{S}) \tag{20}$$

$$\widetilde{\mathrm{PEM}} := \mathrm{PEM}_{\ell,\tilde{N},\tilde{\lambda}}(\tilde{S}) \tag{21}$$

of the individual results of Table 4, the optimal $N$ was determined as $N = 200$. Results obtained with TMAE are omitted for reasons of space.

LS-RBF outperforms all the competing methods for MSE and LAD-RBF for MAE. However, it is important to recall that LS-RBF is based on minimizing MSE and so it LAD-RBF designed to minimize MAE over the training data. TMSE can be described as a more interesting measure, because it is highly robust to outliers. While LTS-RBF is designed to minimize TMSE over the training data, both Tables 4 and 5 show that TLS-RBF is the winner for the evaluation over test sets for TMSE and also for TMAE, where the latter is not shown for reasons of space. Using TMSE, LS-RBF networks are revealed to be harmed by outliers heavily just like in Section 3. LWS-RBF, LWA-RBF, LTS-RBF, and LTA-RBF networks perform quite well although they stay slightly behind TLS-RBF networks. A detailed analysis now presented here reveals the sensitivity of LWS-RBF and LWA-RBF networks to the adjustment of the learning rate. Their loss functions are highly nonlinear and have the form of a minimum obtained across all the individual losses of individual LS-RBF or LAD-RBF networks, respectively.

Using a regularization usually decreases the prediction error here. This is also true for the $L_1$-regularization, which induces sparsity, i. e. the prediction error is improved in spite of a smaller number of relevant parameters used within the network. The results with $L_2$-regularization turn out to be superior to those with $L_1$-regularization. If the variable selection is performed, the prediction becomes slightly weaker; Tables 4 and 5 show the numbers of variables selected by the variable selection. There turns out to be a set of some 30 relevant variables for the Communities and Crime Dataset and 40 to 50 for the Residential Building Dataset, while the remaining variables may be ignored without a heavy influence on the resulting prediction performance.

Linear combinations of two versions of the RBF networks do not yield promising results and are not shown here for reasons of space. For example, a combination of LS-RBF and LWS-RBF was considered, where the weights of both versions for their weighted average was searched for in order to maximize the resulting cross-validation MSE.

## 5. CONCLUSIONS

The experiments presented in this paper can be described as the first available study comparing various robust regularized RBF networks. All the versions of RBF networks that are used in this paper can be formulated also for the context of multilayer perceptrons (MLPs). Particular attention is paid here to the effect of a backward variable selection on the performance of (robust and/or regularized) RBF networks. Using TMSE as a highly robust measures of prediction performance, the best results are obtained for the novel robust TLS-RBF networks with optimal $\tau_1$ and $\tau_2$ obtained according to Algorithm 1. The high vulnerability of standard RBF networks (LS-RBF) with respect to outliers is demonstrated here clearly. The RBF networks based on the LWS, LWA, LTS, and LTA estimators stay slightly behind the best results for TMSE, however their weight functions were chosen as fixed and were not optimized for the sake of additional computational demands. We can also say that $L_2$-regularization turns out to be more suitable compared to $L_1$-regularization here.

| Method | No dimensionality reduction | | | | | | | | | Backward variable selection | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | | | MAE | | | TMSE | | | MSE | | | MAE | | | TMSE | | |
| | Regularization: | | | Regularization: | | | Regularization: | | | Regularization: | | | Regularization: | | | Regularization: | | |
| | - | $L_2$ | $L_1$ | - | $L_2$ | $L_1$ | - | $L_2$ | $L_1$ | - | $L_2$ | $L_1$ | - | $L_2$ | $L_1$ | - | $L_2$ | $L_1$ |
| LS-RBF | 1.71 | 1.50 | 1.64 | 1.25 | 1.07 | 1.18 | 0.46 | 0.33 | 0.41 | 1.95 | 1.80 | 1.86 | 1.39 | 1.13 | 1.18 | 0.40 | 0.35 | 0.37 |
| # of var. | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 36 | 34 | 30 | 40 | 42 | 36 | 43 | 39 | 35 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LAD-RBF | 1.93 | 1.69 | 1.84 | 0.97 | 0.79 | 0.82 | 0.31 | 0.25 | 0.29 | 2.23 | 2.10 | 2.19 | 1.16 | 0.82 | 0.86 | 0.34 | 0.29 | 0.31 |
| # of var. | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 35 | 33 | 31 | 38 | 41 | 32 | 32 | 30 | 29 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| TLS-RBF | 2.47 | 2.30 | 2.36 | 1.20 | 1.03 | 1.14 | 0.19 | 0.16 | 0.19 | 2.78 | 2.51 | 2.64 | 1.25 | 1.14 | 1.18 | 0.23 | 0.19 | 0.21 |
| # of var. | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 31 | 29 | 28 | 35 | 40 | 30 | 28 | 27 | 24 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LTS-RBF | 2.86 | 2.67 | 2.78 | 1.22 | 1.05 | 1.19 | 0.21 | 0.17 | 0.18 | 3.20 | 3.03 | 3.11 | 1.26 | 1.15 | 1.17 | 0.24 | 0.20 | 0.21 |
| # of var. | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 37 | 34 | 35 | 42 | 41 | 32 | 33 | 31 | 30 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LTA-RBF | 2.65 | 2.49 | 2.56 | 1.22 | 1.03 | 1.17 | 0.23 | 0.19 | 0.20 | 2.96 | 2.78 | 2.84 | 1.26 | 1.15 | 1.18 | 0.25 | 0.22 | 0.22 |
| # of var. | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 37 | 34 | 32 | 42 | 41 | 33 | 34 | 31 | 27 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LWS-RBF | 2.92 | 2.75 | 2.84 | 1.24 | 1.06 | 1.14 | 0.20 | 0.17 | 0.19 | 3.24 | 3.05 | 3.13 | 1.27 | 1.14 | 1.17 | 0.25 | 0.20 | 0.21 |
| # of var. | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 30 | 28 | 25 | 42 | 42 | 35 | 29 | 28 | 25 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LWA-RBF | 2.85 | 2.68 | 2.77 | 1.24 | 1.06 | 1.13 | 0.22 | 0.18 | 0.21 | 3.18 | 3.02 | 3.09 | 1.30 | 1.18 | 1.19 | 0.24 | 0.21 | 0.20 |
| # of var. | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 32 | 30 | 30 | 42 | 43 | 34 | 32 | 29 | 24 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| MLP | 1.99 | 1.83 | 1.94 | 1.36 | 1.19 | 1.26 | 0.43 | 0.37 | 0.39 | 2.12 | 1.90 | 1.97 | 1.52 | 1.26 | 1.27 | 0.36 | 0.33 | 0.34 |
| # of var. | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 36 | 34 | 31 | 44 | 45 | 39 | 38 | 34 | 31 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LWS-MLP | 2.77 | 2.61 | 2.70 | 1.23 | 1.04 | 1.14 | 0.26 | 0.22 | 0.23 | 3.06 | 2.88 | 2.94 | 1.33 | 1.30 | 1.31 | 0.29 | 0.26 | 0.27 |
| # of var. | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 33 | 31 | 30 | 41 | 45 | 35 | 31 | 29 | 25 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |

**Tab. 4.** Results for the Communities and Crime Dataset of Section 4 for different regression methods using different regularization types.

| Method | No dimensionality reduction | | | | | | | | | Backward variable selection | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | | | MAE | | | TMSE | | | MSE | | | MAE | | | TMSE | | |
| | Regularization: | | | Regularization: | | | Regularization: | | | Regularization: | | | Regularization: | | | Regularization: | | |
| | - | $L_2$ | $L_1$ | - | $L_2$ | $L_1$ | - | $L_2$ | $L_1$ | - | $L_2$ | $L_1$ | - | $L_2$ | $L_1$ | - | $L_2$ | $L_1$ |
| LS-RBF | 3.91 | 3.60 | 3.86 | 2.72 | 2.33 | 2.58 | 0.74 | 0.52 | 0.58 | 4.32 | 4.07 | 4.29 | 3.05 | 2.76 | 2.89 | 0.81 | 0.58 | 0.62 |
| # of var. | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 45 | 48 | 41 | 47 | 50 | 42 | 52 | 55 | 45 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LAD-RBF | 4.35 | 4.03 | 4.22 | 2.27 | 1.95 | 2.09 | 0.58 | 0.42 | 0.51 | 4.80 | 4.41 | 4.67 | 2.43 | 2.05 | 2.22 | 0.64 | 0.48 | 0.53 |
| # of var. | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 47 | 49 | 39 | 47 | 48 | 37 | 50 | 52 | 44 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| TLS-RBF | 4.45 | 4.14 | 4.30 | 2.31 | 1.99 | 2.13 | 0.54 | 0.40 | 0.46 | 4.85 | 4.44 | 4.68 | 2.51 | 2.16 | 2.23 | 0.60 | 0.44 | 0.49 |
| # of var. | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 41 | 45 | 36 | 42 | 46 | 33 | 45 | 41 | 43 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LTS-RBF | 5.18 | 4.87 | 5.06 | 2.54 | 2.18 | 2.29 | 0.55 | 0.44 | 0.49 | 5.71 | 5.33 | 5.46 | 2.67 | 2.27 | 2.36 | 0.61 | 0.47 | 0.53 |
| # of var. | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 42 | 44 | 34 | 45 | 45 | 37 | 45 | 41 | 31 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LTA-RBF | 5.22 | 4.90 | 5.01 | 2.51 | 2.26 | 2.38 | 0.56 | 0.46 | 0.48 | 5.49 | 5.15 | 5.37 | 2.66 | 2.33 | 2.48 | 0.61 | 0.45 | 0.51 |
| # of var. | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 41 | 43 | 33 | 43 | 45 | 34 | 45 | 43 | 36 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LWS-RBF | 4.67 | 4.34 | 4.57 | 2.69 | 2.36 | 2.47 | 0.56 | 0.44 | 0.49 | 5.21 | 4.86 | 5.03 | 2.67 | 2.48 | 2.57 | 0.62 | 0.48 | 0.52 |
| # of var. | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 38 | 41 | 31 | 43 | 44 | 35 | 44 | 41 | 31 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LWA-RBF | 4.83 | 4.51 | 4.75 | 2.42 | 2.20 | 2.35 | 0.56 | 0.40 | 0.47 | 5.10 | 4.68 | 4.94 | 2.64 | 2.29 | 2.43 | 0.62 | 0.50 | 0.54 |
| # of var. | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 38 | 43 | 31 | 43 | 45 | 33 | 45 | 42 | 32 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| MLP | 4.12 | 3.84 | 4.08 | 2.75 | 2.37 | 2.50 | 0.75 | 0.52 | 0.57 | 4.45 | 4.05 | 4.28 | 3.05 | 2.66 | 2.85 | 0.78 | 0.61 | 0.66 |
| # of var. | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 47 | 50 | 39 | 46 | 51 | 39 | 49 | 53 | 40 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |
| LWS-MLP | 4.52 | 4.19 | 4.35 | 2.56 | 2.21 | 2.40 | 0.59 | 0.46 | 0.53 | 4.85 | 4.43 | 4.55 | 2.73 | 2.45 | 2.58 | 0.62 | 0.49 | 0.55 |
| # of var. | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 105 | 40 | 43 | 35 | 44 | 47 | 37 | 46 | 44 | 36 |
| $\lambda$ | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.1 | 0.1 | - | 0.01 | 0.01 |

**Tab. 5.** Results for the Residential Building Dataset of Section 4 for different regression methods using different regularization types.

Robustness and regularization turn out to be beneficial also for MLPs in accordance with previous results e. g. of [18]. The prior variable selection allows to reduce the number of relevant variables at the cost of a small weakening of the prediction ability. On the whole, robust regularized versions RBF networks turn out to be powerful and computationally straightforward. The backward variable selection may contribute to a more efficient computation of RBF networks [13]. The combination of robustness and regularization may be exploited also in other contexts, particularly in classification tasks, and might be extended to deep networks with RBF layers or for energy-efficient computation of convolutional neural networks (CNNs) [39].

Regularization ensures local robustness to small perturbations e. g. due to rounding. At the same time, regularization may be interpreted as a Bayesian approach, which corresponds to estimating (training) the parameters with a possible contamination by small errors [41]. For this reason, regularized estimation is important for the field of approximate neurocomputing, where the effect of rounding the parameters or computing in a low-precision arithmetic are of interest [5]. As future work, it is intended to study the connection of regularized training with Bayesian estimation also for robust RBF networks; in this context, we perceive the training of robust MLPs as a maximization of certain quasi-likelihood functions, i. e. as an optimization task analogous to likelihood-based statistical estimation problems.

Finally, let us comment the possibility of comparing the results with other analyses of the given data published elsewhere. Although the datasets are benchmarking, i. e. suitable for computing different methods and making comparisons, comparing with different papers would require to use exactly the very same setup that was used in these papers: the same pre-processing, data standardization, details of cross-validation, even the same construction of the test set. Typically, however, the experimental papers do not describe all the details of their analysis, so the results are not directly comparable. For example, the pre-processing applied to the analysis of the residential building dataset is not described in [20]. In addition, sophisticated pre-processing (cleaning the data, transformations) is often performed with the given data. For example in [48], the pre-processing was performed to reduce noise and to reduce the influence of outliers. Our approach is much more realistic: we keep the data as much unchanged as possible, because we want to retain outliers. Thus, in our setup, the disadvantages of standard RBF networks are revealed and the benefits of the robust training are demonstrated.

## ACKNOWLEDGEMENTS

The work is supported by projects 24-10078S (J. Kalina) and 22-02067S (P. Vidnerová and P. Janáček) of the Czech Science Foundation. The authors would like to thank Jiří Tumpach and Jakub Krett for technical help.

(Received August 7, 2023)

## REFERENCES


[1] G. Blokdyk: Artificial Neural Network: A Complete Guide. Createspace Independent Publishing Platform, Scotts Valey 2017.

[2] A. G. Borş and I. Pitas: Robust RBF networks. In: Radial basis function networks 1. (R. J. Howlett, L. C. Jain, and J. Kacprzyk, eds.). Recent developments in theory and applications, Physica Verlag Rudolf Liebing KG, Vienna 2001, pp. 123–133.

[3] K. Boudt, P. J. Rousseeuw, S. Vanduffel, and T. Verdonck: The minimum regularized covariance determinant estimator. Statist. Computing *30* (2020), 113–128. DOI:10.1007/s11222-019-09869-x

[4] S. Chatterjee and A. S. Hadi: Regression Analysis by Example. (Fifth edition.) Wiley, Hoboken 2012.

[5] Y. B. Cheng, X. H. Chen, H. L. Li, Z. Y. Cheng, R. Jiang et al.: Analysis and comparison of Bayesian methods for measurement uncertainty evaluation. Math. Problems Engrg. (2018), 7509046. DOI:10.1155/2018/7509046

[6] F. Chollet: Keras. `https://github.com/fchollet/keras` (2015).

[7] Y. Dodge and J, Jurečková: Adaptive Regression. Springer, New York 2000.

[8] J. Dong, Y. Zhao, C. Liu, Z. F. Han, and C. S. Leung: Orthogonal least squares based center selection for fault-tolerant RBF networks. Neurocomputing *339* (2019), 217–231. DOI:10.1016/j.neucom.2019.02.039

[9] D. Dua and C. Graff: UCI Machine Learning Repository. University of California, Irvine, `http://archive.ics.uci.edu/ml` (2019).

[10] E. Egrioglu, E. Bas, and O. Karahasan: Winsorized dendritic neuron model artificial neural network and a robust training algorithm with Tukey's biweight loss function based on particle swarm optimization. Granular Comput. *8* (2023), 491–501. DOI:10.1007/s41066-022-00345-y

[11] A. H. Fath, F. Madanifar, and M. Abbasi: Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems. Petroleum *6* (2020), 80–91. DOI:10.1016/j.petlm.2018.12.002

[12] M. Hallin, D. Paindaveine, and M. Šiman: Multivariate quantiles and multiple-output regression quantiles: From $L_1$ optimization to halfspace depth. Ann. Statist. *38* (2010), 635–669. DOI:10.1214/09-aos723

[13] I. Han, X. Qian, H. Huang, and T. Huang: Efficient design of multicolumn RBF networks. Neurocomputing *450* (2021), 253–263. DOI:10.1016/j.neucom.2021.04.040

[14] T. Hastie, R. Tibshirani, and J. Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. (Second edition.). Springer, New York 2009.

[15] J. Jurečková, J. Picek, and M. Schindler: Robust Statistical Methods with R. (Second edition.) Chapman and Hall/CRC, Boca Raton 2019.

[16] J. Kalina and J. Tichavský: On robust estimation of error variance in (highly) robust regression. Measurement Sci. Rev. *20* (2020), 6–14. DOI:10.2478/msr-2020-0002

[17] J. Kalina, A. Neoral, and P. Vidnerová: Effective automatic method selection for nonlinear regression modeling. Int. J. Neural Syst. *31* (2021), 2150020. DOI:10.1142/S0129065721500209

[18] J. Kalina, J. Tumpach, and M. Holeňa: On combining robustness and regularization in training multilayer perceptrons over small data. In: 2000 International Joint Conference on Neural Networks (IJCNN), IEEE 2022. DOI:10.1109/ijcnn55064.2022.9892510

[19] M. E. Karar: Robust RBF neural network-based backstepping controller for implantable cardiac pacemakers. Int. J. Adaptive Control Signal Process. *32* (2018), 1040–1051. DOI:10.1002/acs.2884

[20] I. A. Khan, T. Hussain, A. Ullah, S. Rho, M. Lee, M., and S. W. Baik: Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid CNN with a LSTM-AE based framework. Sensors *20* (2020), 1399. DOI:10.3390/s20051399

[21] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter: Self-normalizing neural networks. In: NIPS'17: Proc. 31st International Conference on Neural Information Processing Systems, Curran Associates, New York 2017, pp. 972–981.

[22] M. A. Knefati, P. E. Chauvet, S. N'Guyen, and B. Daya: Reference curves estimation using conditional quantile and radial basis function network with mass constraint. Neural Process. Lett. *43* (2016), 17–30. DOI:10.1007/s11063-014-9399-9

[23] R. Koenker: Quantile regression: 40 years on. Annual Rev. Econom. *9* (2917), 155–176. DOI:10.1146/annurev-economics-063016-103651

[24] M. Kordos and A. Rusiecki: Reducing noise impact on MLP training — Techniques and algorithms to provide noise-robustness in MLP network training. Soft Comput. *20* (2016), 46–65. DOI:10.1007/s00500-015-1690-9

[25] C. C. Lee, P. C. Chung, J. R. Tsai, and C. I. Chang: Robust radial basis function neural networks. IEEE Trans. Systems Man Cybernet. *B 29* (1999), 674–685. DOI:10.1109/3477.809023

[26] X. Li and Y. Sun: Application of RBF neural network optimal segmentation algorithm in credit rating. Neural Computing Appl. *33* (2021), 8227–8235. DOI:10.1007/s00521-020-04958-9

[27] Z. Liu, C. S. Leung, and H. C. So: Formal convergence analysis on deterministic $\ell_1$-regularization based mini-batch learning for RBF networks. Neurocomputing *532* (2023), 77–93. DOI:10.1016/j.neucom.2023.02.012

[28] National Institute of Standards and Technology (NIST): Nonlinear Regression Datasets. `https://www.itl.nist.gov/div898/strd/nls/nls_main.shtml` (2022).

[29] C. Paul and G. K. Vishwakarma: Back propagation neural networks and multiple regressions in the case of heteroscedasticity. Commun. Statist. Simul. Comput. *46* (2017), 6772–6789. DOI:10.1080/03610918.2016.1212066

[30] G. Petneházi: Quantile convolutional neural networks for value at risk forecasting. Machine Learning Appl. *6* (2021), 100096. DOI:10.1016/j.mlwa.2021.100096

[31] T. Poggio and S. Smale: The mathematics of learning: Dealing with data. Notices Amer. Math. Soc. *50* (2003), 537–544. DOI:10.1109/icnnb.2005.1614546

[32] B. Procházka: Regression quantiles and trimmed least squares estimator in the nonlinear regression model. Comput. Statist. Data Anal. *6* (1988), 385–391. DOI:10.1016/0167-9473(88)90078-3

[33] Q. Que and M. Belkin: Back to the future: Radial basis function network revisited. IEEE Trans. Pattern Anal. Machine Intell. *42* (2020), 1856–1867. DOI:10.1109/TPAMI.2019.2906594

[34] Y. Romano, E. Patterson, and E. J. Candès: Conformalized quantile regression. ArXiv 2019. https://arxiv.org/abs/1905.03222

[35] A. Rusiecki: Trimmed categorical cross-entropy for deep learning with label noise. Electron. Lett. *55* (2019), 319–320. DOI:10.1049/el.2018.7980

[36] A. Rusiecki: Robust learning algorithm based on LTA estimator. Neurocomputing *120* (2013), 624–632. DOI:10.1016/j.neucom.2013.04.008

[37] A. K. M. E. Saleh, J. Picek, and J. Kalina: R-estimation of the parameters of a multiple regression model with measurement errors. Metrika *75* (2012), 311–328. DOI:10.1007/s00184-010-0328-2

[38] A. K. Seghouane and N. Shokouhi: Adaptive learning for robust radial basis function networks. IEEE Trans. Cybernet. *51* (2021), 2847–2856. DOI:10.1109/TCYB.2019.2951811

[39] J. Šíma, P. Vidnerová, and V. Mrázek: Energy complexity model for convolutional neural networks. Lecture Notes Computer Sci. *14263* (2023), 186–198. DOI:10.1016/j.comcom.2022.10.029

[40] M. J. Su and W. Deng: A fast robust learning algorithm for RBF network against outliers. Lecture Notes Computer Sci. *4113* (2006), 280–285. DOI:10.1007/11816157_28

[41] V. Sze, Y. B. Chen, T. J. Yang, and J. S. Emer: Efficient Processing of Deep Neural Networks. Morgan and Claypool, San Rafael 2020.

[42] J. Tukey: Comparing individual means in the analysis of variance. Biometrics *5* (1949), 99–114. DOI:10.2307/3001913

[43] I. Ullah, H. Y. Youn, and Y. H. Han: An efficient data aggregation and outlier detection scheme based on radial basis function neural network for WSN. J. Ambient Intell. Humanized Comput., in press (2022).

[44] J.Á. Víšek: Consistency of the least weighted squares under heteroscedasticity. Kybernetika *47* (2011), 179–206.

[45] T. Werner: Quantitative robustness of instance ranking problems. ArXiv 2021. https://arxiv.org/abs/2103.07198 v2.

[46] R. R. Wilcox: Introduction to Robust Estimation and Hypothesis Testing. Fourth edition. Academic Pres, London 2017.

[47] C. Yang, S. K. Oh, W. Pedrycz, Z. Fu, and B. Yang: Design of reinforced fuzzy radial basis function neural network classifier driven with the aid of iterative learning techniques and support vector-based clustering. IEEE Trans. Fuzzy Systems *29* (2021), 2506–2520. DOI:10.1109/TFUZZ.2020.3001740

[48] P. Yerpude and V. Gudur: Predictive modelling of crime dataset using data mining. Int. J. Data Mining Knowledge Management Process *7* (2017), 43–58. DOI:10.5121/ijdkp.2017.7404

[49] D. Zhang, G. Zang, J. Li, K. Ma, and H. Liu: Prediction of soybean price in China using QR-RBF neural network model. Computers Electron. Agriculture *154* (2018), 10–17. DOI:10.1016/j.compag.2018.08.016

[50] Y. Zuo: New algorithms for computing the least trimmed squares estimator. ArXiv 2022. https://arxiv.org/abs/2203.10387

*Jan Kalina, The Czech Academy of Sciences, Institute of Computer Science, Pod Vodárenskou věží 2, 182 00 Praha 8, Czech Republic and The Czech Academy of Sciences, Institute of Information Theory and Automation, Pod Vodárenskou věží 4, 182 00 Praha 8. Czech Republic.*
*e-mail: kalina@cs.cas.cz*

*Petra Vidnerová, The Czech Academy of Sciences, Institute of Computer Science, Pod Vodárenskou věží 2, 182 00 Praha 8. Czech Republic.*
*e-mail: petra@cs.cas.cz*

*Patrik Janáček, The Czech Academy of Sciences, Institute of Computer Science, Pod Vodárenskou věží 2, 182 00 Praha 8. Czech Republic.*
*e-mail: patrik.janacek@email.cz*