Robust Echo State Network for Recursive System Identification*

Renan Bessa and Guilherme A. Barreto

Department of Teleinformatics Engineering, Federal University of Ceará
Center of Technology, Campus of Pici, Fortaleza, Ceará, Brazil
renanbessa.etec@gmail.com, gbarreto@ufc.br

Abstract. The use of recurrent neural networks in online system identification is very limited in real-world applications, mainly due to the propagation of errors caused by the iterative nature of the prediction task over multiple steps ahead. Bearing this in mind, in this paper, we revisit design issues regarding the robustness of the *echo state network* (ESN) model in such online learning scenarios using a recursive estimation algorithm and an outlier robust-variant of it. By means of a comprehensive set of experiments, we show that the performance of the ESN is dependent on the adequate choice of the feedback pathways and that the prediction instability is amplified by the norm of the output weight vector, an often neglected issue in related studies.

Keywords: Online system identification · Recurrent neural networks · Echo state network · Recursive estimation · Robustness to outliers.

1 Introduction

The echo state network (ESN) [7] is a recurrent neural network (RNN) that has a large set of neurons, the so-called reservoir, with sparse interconnections and feedback pathways. The input weights of the reservoir neurons, the internal and the ones responsible for connecting the system input and output, are fixed and randomly assigned. Training of this network requires the estimation of the weights of the neurons in the output layer (aka, readout layer). This estimation is carried out via linear regression, usually by means of the well-known OLS method. The randomized nature of the ESN combined with the linear estimation of the output layer weights makes its design very simple if compared to the other RNNs. In Figure 1 is depicted the standard architecture of the ESN.

* This study was financed by the following Brazilian research funding agencies: CAPES (finance code 001), FUNCAP (BMD-008-01413.01.02-17) and CNPq (grant 309451/2015-9).

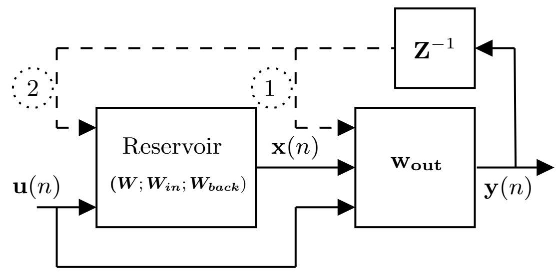


Fig. 1: The basic setup of ESN indicating feedforward (solid lines) and feedback (dashed lines) pathways. $\mathbf{W}$, $\mathbf{W}_{in}$, $\mathbf{W}_{back}$ and $\mathbf{W}_{out}$ are weight matrices.

The ESN has been used as a powerful tool for the prediction of chaotic series, attractor reconstruction, and for nonlinear system modelling in general [8]. In order to improve this network to handle real-world data, which commonly contain non-Gaussian noise and inconsistent points (outliers), some works introduced robust estimation methods for the estimation of the readout layer weights. One of the simplest methods is the Tikhonov regularization, which penalizes weight vector with a high norm, reducing model overfitting to the corrupted data [10].

More complex approaches have also been proposed, including Bayesian inference, replacing the usual Gaussian likelihood function, which is very sensitive to the outliers, by a Laplacian [9] or mixed-Gaussian distribution [6]. Furthermore, the use of performance criteria based on the information theoretic learning, called Maximum Correntropy [5] and Generalized Correntropy [13], have also achieved good results in diminishing sensitivity to outliers.

Other robust ESN approaches, replace the usual linear output layer with a nonlinear framework, trying to take full advantage of the dynamics of the reservoir, whose output signals are inputted to the readout layer. One of such approaches applies SVM formulation with robust cost functions such as ϵ-insensitive loss or Huber [11]. Other approaches use kernel adaptive filtering methods, such as the kernel recursive least squares (RLS) algorithm [14]. Although it has not been evaluated directly, there are references that in arrangements of the ESN with Laplacian Eigenmaps algorithm [6], mediating a decrease of the dimensionality of the reservoir states, and with Gaussian process regression models [3] may be able to decrease the sensitivity to outliers.

Despite the powerful modeling capabilities of the aforementioned ESN architectures, their application to nonlinear system identification is limited. This is particularly true for real-world application scenarios involving online long-term iterative predictions over multiple steps ahead. Since the chance to meet outliers in such scenarios is very high, the propagation of errors due to the iterative nature of the prediction task causes divergence (i.e. instability) in the predicted signals as time passes. Such instability is amplified by the norm of the output weight vector, an often neglected issue in previous studies.

Bearing the stability of long-term prediction in mind, in this work we revisit the RLS-ESN [8] model, ESN combined with the RLS estimation algorithm, for online system identification in the presence of outliers. By exploring variations in the feedback pathways of the ESN and replacing the RLS algorithm with a robust variant of it [2] we show that the ESN can be safely used for long-term multiple-step-ahead prediction tasks. The role of the norm of the weights in the propagation of errors is also evaluated. A comprehensive set of computer simulations using data from real-world and synthetic systems are carried out, which are contaminated by outliers in different proportions.

2 Fundamentals of the Echo State Network

The parameter estimation process in discrete-time system identification requires the availability of input and output observations from a system of interest: $(\mathbf{u}(n), \mathbf{t}(n))_{n=0}^{N_{all}}$, where $\mathbf{u}(n) = [u_1(n),\ u_2(n),\ \ldots,\ u_K(n)]^T$ is a K-dimensional input vector and $\mathbf{t}(n) = [t_1(n),\ t_2(n),\ \ldots,\ t_L(n)]^T$ is an L-dimensional output vector at a given instant n, $n = 1, \ldots, N_{all}$. This dataset is then divided into training and test subsets, $N_{all} = N_{train} + N_{test}$. Once adequate training is completed, the ESN is required to predict output vectors $\mathbf{y}(n) \in \mathbb{R}^{L\times 1}$ with the smallest possible deviation from $\mathbf{t}(n)$.

The activations of the R reservoir neurons, also called state variables, are denoted by $\mathbf{x}(n) \in \mathbb{R}^{R\times 1}$, being computed as

$$\mathbf{x}(n) = \mathbf{f}(\mathbf{W}\mathbf{x}(n-1) + \mathbf{W}_{in}\mathbf{u}(n) + \mathbf{W}_{back}(\mathbf{t}(n-1) + \mathbf{v}_1(n)) + \mathbf{v}_2(n) - \mathbf{b}), \quad (1)$$

where $\mathbf{f}(\cdot)$ is a nonlinear activation function with element-wise operation, usually the logistic sigmoid or the tanh function, $\mathbf{W} \in \mathbb{R}^{R\times R}$ is the internal weight matrix of the reservoir, which is responsible for the reservoir feedback pathways, and $\mathbf{W}_{in} \in \mathbb{R}^{R\times K}$ and $\mathbf{W}_{back} \in \mathbb{R}^{R\times L}$ are the weight matrices that connect the input and output of the model to the reservoir, respectively. The terms $\mathbf{v}_1(n) \in \mathbb{R}^{L\times 1}$ and $\mathbf{v}_2(n) \in \mathbb{R}^{R\times 1}$ are optional white noise vectors, while $\mathbf{b} \in \mathbb{R}^{R\times 1}$ is the bias vector.

The output of the neural model is computed as

$$\mathbf{y}(n) = \mathbf{f}_{out}\left(\mathbf{W}_{out}\mathbf{h}(n)\right), \quad (2)$$

where $\mathbf{f}_{out}(\cdot)$ is the activation function of the output neurons (usually, the identity function), $\mathbf{W}_{out} \in \mathbb{R}^{L\times(1+K+R+L)}$ is the output weight matrix, $\mathbf{h}(n) = [-1,\ \mathbf{u}(n),\ \mathbf{x}(n),\ (\mathbf{t}(n-1) + \mathbf{v}_1(n))]^T \in \mathbb{R}^{H\times 1}$ is a concatenated vector, so that $H = 1 + K + R + L$.

The elements of the weight matrices $\mathbf{W}$, $\mathbf{W}_{in}$ and $\mathbf{W}_{back}$ are randomly assigned and kept fixed during training and testing of the model. However, $\mathbf{W}$ must be sparse, i.e. it must contain only a small percentage of nonzero elements. In order to guarantee the "echo state" property[1], the spectral radius of $\mathbf{W}$ must be chosen to be within the unit circle. The magnitudes of the noise vectors $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$ must be chosen aiming at reaching a tradeoff between the stability and accuracy of the model. These vectors also help to regularize the solution. Details on the ESN configuration can be found in [7].

[1] That of relating asymptotic properties of the excited reservoir dynamics to the driving signal.

2.1 Recursive algorithms for parameter estimation

In this paper we are dealing with SISO systems; thus, a single output is considered. The goal of training is to estimate recursively the weight vector $\mathbf{w}_{out} \in \mathbb{R}^{H\times 1}$. We aim at comparing two estimation methods, namely: the standard RLS algorithm, whose cost function is $J_{LS}(n) = \sum_{i=1}^{n} \gamma^{n-i} e^2(i)$, where $0 < \gamma \leq 1$ is the forgetting factor and $e(n) = t(n) - \mathbf{w}_{out}^T(n-1)\mathbf{h}(n)$ is the prediction error at instant n, and the *recursive Least M-estimate algorithm* (RLM), whose cost function is given by

$$J_\rho(n) = \sum_{i=1}^{n} \gamma^{n-i} \rho(e(i)), \tag{3}$$

where $\rho(e)$ is a function whose purpose is to limit the negative effect caused by very large errors (either caused by non-Gaussian noise or outliers). The RLM algorithm reduces to the standard RLS when $\rho(e(i)) = e^2(i)$.

The optimal output weights can be determined by differentiating $J_\rho(n)$ with respect to $\mathbf{w}_{out}$ and setting the derivatives to zero. After some algebric manipulation, the resulting RLM algorithm involves the following equations:

$$\mathbf{w}_{out}(n) = \mathbf{w}_{out}(n-1) + e(n)\mathbf{k}(n), \tag{4}$$

$$\mathbf{k}(n) = \frac{q(e(n))\mathbf{S}(n-1)\mathbf{h}(n)}{q(e(n))\mathbf{h}^T(n)\mathbf{S}(n-1)\mathbf{h}(n) + \gamma} \tag{5}$$

$$\mathbf{S}(n) = \gamma^{-1}\left(\mathbf{I} - \mathbf{k}(n)\mathbf{h}^T(n)\right)\mathbf{S}(n-1) \tag{6}$$

where $\mathbf{k}(n)$ is the gain vector and $q(e) = \frac{1}{e}\frac{\partial \rho(e)}{\partial e}$. The matrix $\mathbf{S}(n)$ is the online estimate of the inverse of the correlation matrix, i.e. $\mathbf{S}(n) = \mathbf{R}^{-1}(n)$.

In this paper, we use the classic Huber function

$$\rho(e) = \begin{cases} \dfrac{e^2}{2}, & |e| < \xi \\ \\ \xi|e| - \dfrac{\xi^2}{2}, & |e| > \xi, \end{cases} \quad \Rightarrow \quad q(e) = \begin{cases} 1, & |e| < \xi \\ \\ \dfrac{\xi}{|e|}, & |e| > \xi, \end{cases} \tag{7}$$

where ξ is a threshold value. Errors high than this threshold are to be considered an outlier. It should be noted that $q(e) = 1$ for errors smaller than the threshold ξ; otherwise, $q(e) \to 0$ as $|e| \to \infty$.

It is possible to estimate ξ continuously for each new input. For this purpose, one can use the following expression:

$$\xi(n) = 2.576\hat{\sigma}(n), \tag{8}$$

where $\hat{\sigma}(n)$ is a robust estimate of the standard deviation of the residuals:

$$\hat{\sigma}^2(n) = c_1 \text{med}\{e^2(n), ..., e^2(n - N_w + 1)\}, \tag{9}$$

so that $c_1 = 1.483(1 + 5/(N_w - 1))$, med$\{\cdot\}$ is the sample median and N_w is the window length. The robustness and computational complexity of the method increases with increasing the value of N_w.

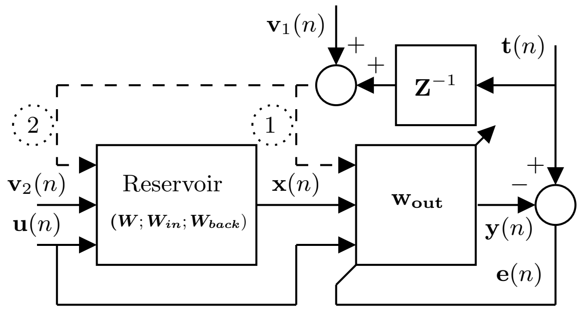


Fig. 2: ESN architecture during training (dashed lines are optional connections).

3 Methodology of Evaluation and Simulation

The ESN architecture diagram is outlined in Figure 2. Our goal is to evaluate the following variants of the ESN architecture: (i) **Model 1** (M1) - connection pathways 1 and 2 disabled; (ii) **Model 2** (M2) - connection pathway 1 enabled, connection pathway 2 disabled; and (iii) **Model 3** (M3) - connection pathways 1 and 2 enabled.

Three approaches to estimating the output weight vector $\mathbf{w}_{out}$ are also evaluated. The standard RLS algorithm is used as a baseline of reference against which the plain RLM algorithm is contrasted. A third approach is specific to the task of iterated prediction, aiming at providing higher stability of the prediciton over longer time horizons, when outliers are present in the data.

RLM with Outlier Detection (RLM-OD): At iteration n, the output prediction is computed as $y(n) = \mathbf{w}_{out}^T(n-1)\mathbf{h}(n)$. The prediction error is computed as $e(n) = t(n) - \mathbf{w}_{out}^T(n-1)\mathbf{h}(n)$ and used by the RLM algorithm as shown in Eqs. (4) and (5). However, if $|e(n)| > \xi$, this has happened because the target output $t(n)$ is probably an outlier. Then, in the next training iteration, we replace the actual observation $t(n)$ with its predicted value $y(n)$ in order to avoid filling in the input regression vector with outliers.

3.1 Evaluation and Simulation

All the datasets used in the experiments come from SISO systems. They are listed in Table 1, with some details added. For the sake of model building and validation, the datasets are divided as follows: the first half of the samples for training, from 0 to $N_{train} - 1$, and the second half for the test, from N_{train} to $N_{all} - 1$. It is worth mentioning that once the systems of interest are SISO, the input vector $\mathbf{u} \in \mathbb{R}^K$, the target vector $\mathbf{t} \in \mathbb{R}^L$, and the random noise vector $\mathbf{v}_1 \in \mathbb{R}^K$ become scalars (i.e. $K = 1$ and $L = 1$). The random noise vector $\mathbf{v}_2 \in \mathbb{R}^R$ has the same dimension of the number of units in the reservoir.

System	Description	# samples (N_{all})	Ref.
TOP	A Synthetic Tenth-Order Problem (Example 5)	2000	[1]
Tank	Cascaded tanks (Tank2.mat; lower tank: $y0$)	7500	[12]
Dryer	A laboratory setup acting like a hair dryer (96-006)	1000	[4]
Exchanger	A liquid-saturated steam heat exchanger (97-002)	4000	[4]
Robot Arm	A flexible robot arm (96-009)	1024	[4]

Table 1: List of datasets used in the computer experiments.

Two criteria were used for the evaluation of the ESN models. The root mean square error (RMSE) of the iterated predictions,

$$RMSE = \sqrt{\frac{\sum_{n=N_{train}}^{N_{all}-1}(t(n) - y(n))^2}{N_{test}}} \tag{10}$$

and the Euclidean norm (l_2) of the output weight vector, $||\mathbf{w}_{out}||$. These two figures of merit are important in assessing the model quality.

The default setting for all simulations of this work is shown in the Table 2. The number of reservoir neurons, R, is checked for each system, without the presence of outliers and with the RLS, seeking a balance between the RMSE, overfitting and stability, for $R \in \{10, 25, 50, 100, 150, ..., 500\}$. The RLM algorithm has an extra parameter, N_w, which defines the size of the sample window for calculating the threshold ξ. We decided to use all available samples up to the iteration n, that is, N_w is increasing and ranges from 0 to $N_{train} - 1$.

The robustness of the methods are evaluated for outliers contamination scenarios of 5%, 10% e 15% of training samples. These are generated by $\sigma_{train}\mathcal{T}(0, 2)$, where σ_{train} is the standard deviation of the original training data and $\mathcal{T}(0, 2)$ is a Student-T distribution with zero mean and two degrees of freedom. The values returned from the Student-T distribution are saturated at ± 20. The systems' inputs and outputs, with and without outliers, are normalized by subtracting the mean and dividing by three times the standard deviation value of the training samples. The RMSE is calculated with the non-normalized data values. Twenty (20) independent training/testing runs are executed for each contamination scenario, for which all weights/biases and outliers are randomly initialized. In each runs, the same weights and the same contamination of the data are applied to each investigated model in order to carry out a fair performance comparison for the different ESM models (M1, M2 and M3) and the parameter estimation methods (RLS, RLM and RLM-OD).

4 Results

Before running the final simulations, the number of neurons in the reservoir was chosen by experimentation with outlier-free data. It was observed the convergence to a minimum value of the RMSE with the increase in the number of

ESN CONFIGURATION	
Spectral Radius ($\mathbf{W}$)	0.98
Sparsity ($\mathbf{W}$)	2%
Input Weights	$\mathbf{W}_{in} \sim U(-0.1, 0.1)$
Feedback Weights	$\mathbf{W}_{back} \sim U(-0.1, 0.1)$
Bias Weights	$\mathbf{b} \sim U(-0.1, 0.1)$
Noise Vector 1	$\mathbf{v}_1 \sim U(-0.0001, 0.0001)$
Noise Vector 2	$\mathbf{v}_2 \sim U(-0.0001, 0.0001)$
RLS/RLM CONFIGURATION	
Forgetting Factor (γ)	0.99998
$\mathbf{S}(0)$	$10^4 I$

Table 2: Default setting of several hyperparameters for the simulations.

Datasets (value of R)		**Model M1**	**Model M2**	**Model M3**
TOP (500)	RMSE	0.0312±0.0105(0.0279)	0.0314±0.0110(0.0270)	0.0344±0.0079(0.0316)
	Norm	98.04±20.34(101.73)	75.22±15.10(76.94)	62.27±16.72(67.88)
Tank (500)	RMSE	0.3227±0.0916(0.3202)	0.1568±0.0244(0.1579)	0.1056±0.0122(0.1033)
	Norm	152.64±47.61(146.76)	5.09±0.82(5.33)	2.36±0.08(2.36)
Dryer (250)	RMSE	0.1356±0.0377(0.1229)	0.1238±0.0212(0.1165)	0.1299±0.0170(0.1237)
	Norm	18.79±2.33(19.01)	8.59±0.85(8.43)	8.34±1.24(8.17)
Exchanger (450)	RMSE	0.2432±0.0305(0.2339)	0.2507±0.0327(0.2400)	0.2312±0.0348(0.2227)
	Norm	37.05±4.28(37.24)	33.449±2.80(33.23)	24.03±2.86(24.74)
Robot Arm (500)	RMSE	0.2840±0.0459(0.2785)	0.2991±0.0552(0.2907)	0.0238±0.0136(0.0198)
	Norm	165.89±16.35(172.85)	118.25±11.39(121.78)	2.33±0.30(2.25)

Table 3: Models' performances using the RLS algorithm with outlier-free data.

neurons for all datasets. In Table 3, it is shown the chosen number of neurons chosen and the results achieved by the RLS-ESN for the three evaluated architectures (M1, M2 and M3) using the RLS algorithm. These results are expressed in the following format: $mean \pm std(median)$, respectively, the mean, standard deviation and median of the RMSE values and the norm of $\mathbf{w}_{out}$ averaged along the 20 independent runs.

Keeping the same number of neurons in the reservoir, the models' performances are better illustrated in Figures 3 and 4, where they can be compared with respect to the different contamination scenarios and to the choice of the robust estimator (RLM and RLM-OD). It should be noted that for the M1 model, the RLM-OD algorithm behaves exactly like the standard RLM, since the output is not fed back (i.e. connections pathways 1 and 2 are disabled).

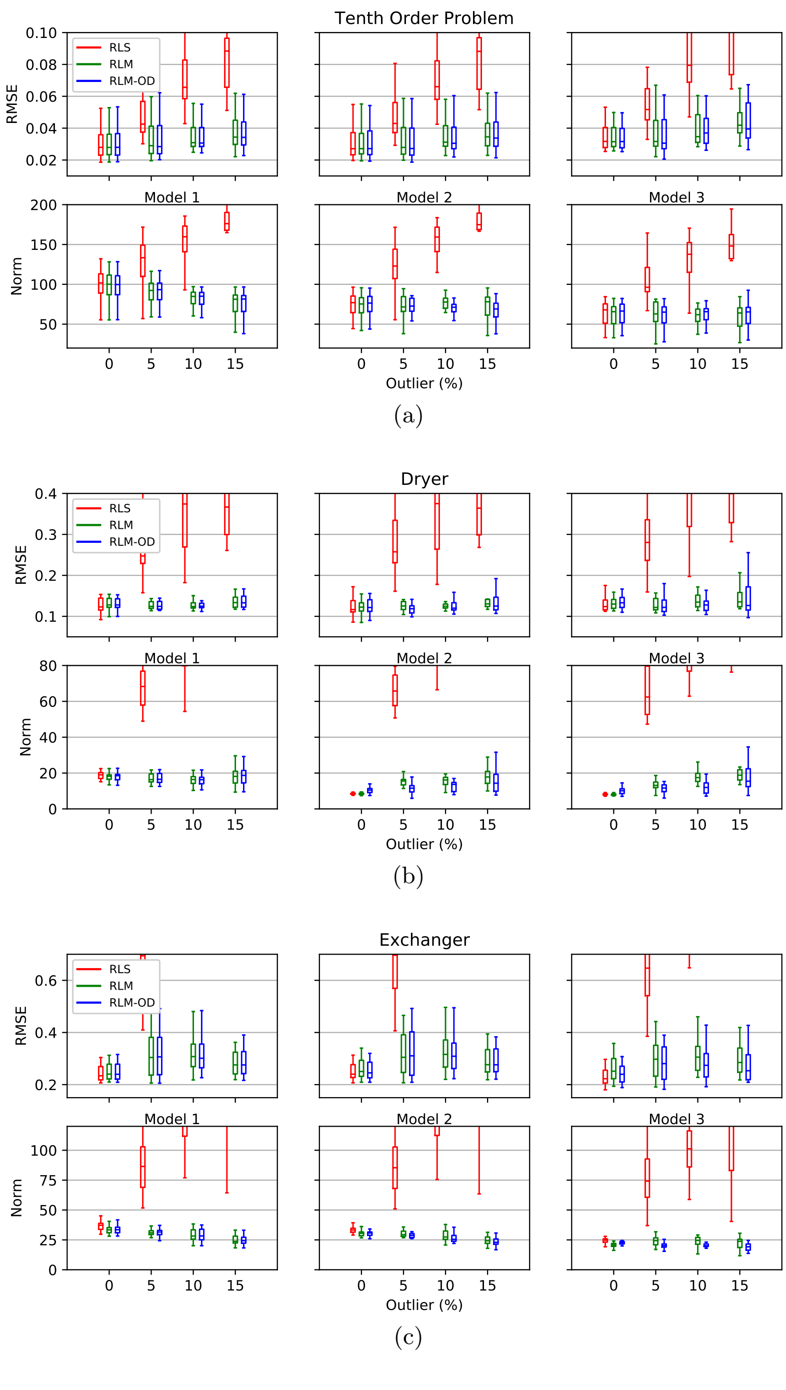
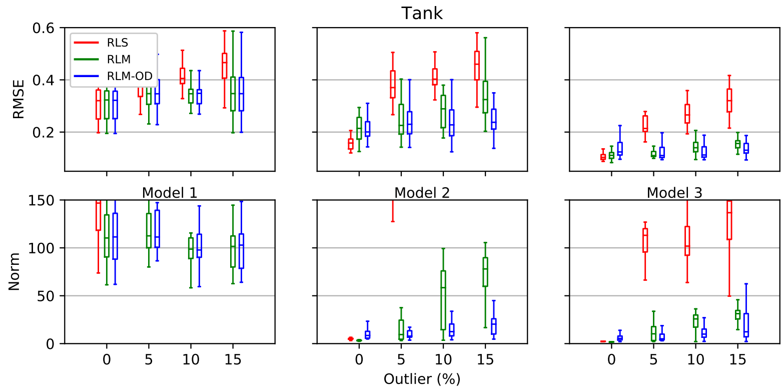
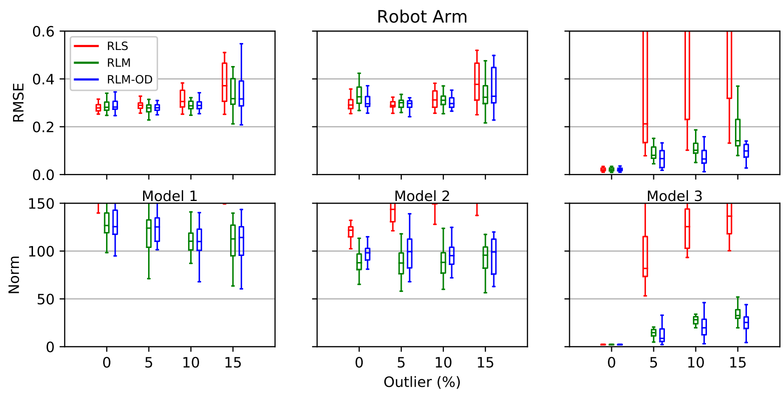


Fig. 3: Performances on the (a) 10-th order problem, (b) Dryer and (c) Exchanger datasets.



(a)



(b)

Fig. 4: Performances on the (a) Tank and (b) Robot arm datasets.

As can be seen in Table 3, the presence of feedback pathways (Models M2 e M3) decreased the norm of the ESN output layer, which leads to an improvement in generalization and avoids overfitting. In other words, the feedback pathways acted as a model regularizer. The M3 model, which feeds the predicted output back to the readout layer and the reservoir, has the smallest norms and the corresponding RMSE values are smaller or very close compared to the other evaluated models.

In Figure 3 we report the results for different contamination scenarios in which the three models presented similar RMSE values when using the RLM-type estimation algorithm. In other words, for these datasets, the M2 and M3 models presented similar performance independently of the RLM-type estimation algorithm used. In terms of the norm of the output vector, all models (M1, M2 and M3) using RLM-like estimation algorithms are practically insensitive

		RLS	RLM	RLM-OD
Tank	RMSE	0.3553	0.1796	0.1063
	Norm	108.46	25.53	8.40
Robot Arm	RMSE	0.2509	0.0997	0.0482
	Norm	108.75	46.09	13.14

Table 4: Numerical results for the *Tank* and *Robot Arm* datasets achieved by the M3 model with a 15% contamination scenario.

to the increase in the amount of outliers. As expected, when the models used the standard RLS algorithm, the norm of the output weight vector increased considerably with the increase in the amount of outliers.

For the datasets Tank and Robot Arm the superior performance of the M3 model over the other two models is evident, as shown in Figure 4. For these datasets, the performance of the M3 model using the proposed RLM-OD estimation algorithm is consistently superior to that of the M3 model using the standard RLM algorithm. This is true in terms of smaller RMSE values, but also in terms of smaller norms for the output weight vector.

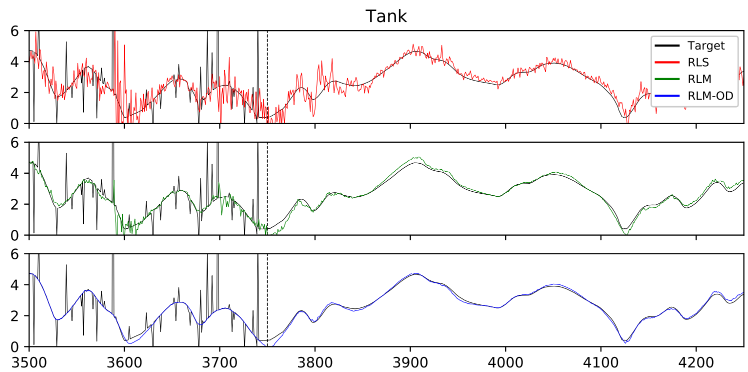
To illustrate the importance of the combined use of robust estimation methods and small norms of the output weight vector in the long-term performance of the M3 model, we report in Figure 5 typical predicted time series for a single training/testing run and a scenario with 15% of outlier contamination. The vertical dotted lines indicate the end of training phase and beginning of the iterated long-term prediction task. The superior performance of the fully recurrent M3 model using the robust estimation algorithms (RLM and RLM-OD) is clear.

These models (M3-RLM and M3-RLM-OD) were both able to learn the underlying dynamics of the system of interest even in the presence of outliers during the training phase and capable of diminishing the effect of error propagation during the testing phase due to the smaller norms of the output weight vector. The corresponding numerical results for this example are reported in Table 5.

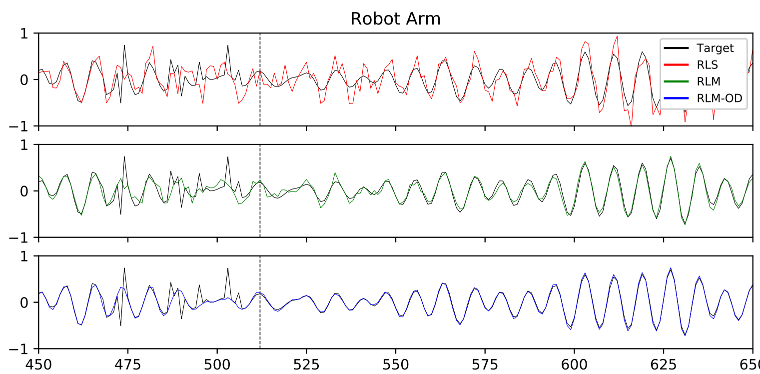
5 Conclusions and Further Work

In this paper, we evaluated the performance of the ESN model for recursive identification task under outlier-contaminated scenarios. Two outlier-robust variants of the RLS algorithm were used for estimating the output weight vector, namely, the RLM algorithm and the proposed RLM-OD algorithm.

It was verified by a comprehensive set of computer experiments using benchmarking datasets that the combined used of feedback pathways and robust estimation algorithms led to regularized recurrent neural network models. Among the evaluated models, the M3-RLM-OD model consistently presented very promising results, being able to keep the norm of the output weight vector small and, hence, reduce the negative influence of outliers in the long term iterated prediction performance of the ESN model.



(a)



(b)

Fig. 5: Typical predicted time series for the (a) *Tank* and (b) *Robot Arm* datasets achieved by the M3 model with a 15% contamination scenario.

Currently, we are extending the experiments carried out in this paper to the online identification of multiple-input, multiple output (MIMO) dynamic systems.

References

1. Atiya, A.F., Parlos, A.G.: New results on recurrent network training: unifying the algorithms and accelerating convergence. IEEE transactions on neural networks **11**(3), 697–709 (2000)
2. Chan, S.C., Zou, Y.X.: A recursive least m-estimate algorithm for robust adaptive filtering in impulsive noise: fast algorithm and convergence performance analysis. IEEE Transactions on Signal Processing **52**(4), 975–991 (2004)

3. Chatzis, S.P., Demiris, Y.: Echo state gaussian process. IEEE Transactions on Neural Networks **22**(9), 1435–1445 (2011)
4. De Moor, B.L.R.: Daisy: Database for the identification of systems. Department of Electrical Engineering, ESAT/SISTA, K.U.Leuven, Belgium http://www.esat.kuleuven.ac.be/sista/daisy/ (accessed January 15, 2019), [Used dataset: Dryer: (96-006); Exchanger (97-002); Robot arm (96-009).]
5. Guo, Y., Wang, F., Chen, B., Xin, J.: Robust echo state networks based on correntropy induced loss function. Neurocomputing **267**, 295–303 (2017)
6. Han, M., Xu, M.: Laplacian echo state network for multivariate time series prediction. IEEE transactions on neural networks and learning systems **29**(1), 238–244 (2018)
7. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report **148**(34), 13 (2001)
8. Jaeger, H.: Adaptive nonlinear system identification with echo state networks. In: Advances in neural information processing systems. pp. 609–616 (2003)
9. Li, D., Han, M., Wang, J.: Chaotic time series prediction based on a novel robust echo state network. IEEE Transactions on Neural Networks and Learning Systems **23**(5), 787–799 (2012)
10. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Computer Science Review **3**(3), 127–149 (2009)
11. Shi, Z., Han, M.: Support vector echo-state machine for chaotic time-series prediction. IEEE Transactions on Neural Networks **18**(2), 359–372 (2007)
12. Wigren, T.: Input-output data sets for development and benchmarking in nonlinear identification. Technical Reports from the department of Information Technology **20**, 2010–020 (2010)
13. Zhang, C., Guo, Y., Wang, F., Chen, B.: Generalized maximum correntropy-based echo state network for robust nonlinear system identification. In: 2018 International Joint Conference on Neural Networks (IJCNN). pp. 1–6. IEEE (2018)
14. Zhou, H., Huang, J., Lu, F., Thiyagalingam, J., Kirubarajan, T.: Echo state kernel recursive least squares algorithm for machine condition prediction. Mechanical Systems and Signal Processing **111**, 68–86 (2018)