Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-5 (2016) 031–036

A comparison of fuzzy identification methods on benchmark datasets

Darko Aleksovski * Dejan Dovžan ** Sašo Džeroski * Juš Kocijan ***

* Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, Ljubljana, Slovenia
** Faculty of Electrical Engineering, University of Ljubljana, Tržaška cesta 25, Ljubljana, Slovenia
*** Department of Systems and Control, Jožef Stefan Institute, Jamova cesta 39, Ljubljana, Slovenia, and University of Nova Gorica, Vipavska 13, Nova Gorica, Slovenia
(e-mail: jus.kocijan@ijs.si)

Abstract: In this paper, we address the task of discrete-time modeling of nonlinear dynamic systems. We use Takagi-Sugeno fuzzy models built by LOLIMOT and SUHICLUST, as well as ensembles of LOLIMOT fuzzy models to accurately model nonlinear dynamic systems from input-output data. We evaluate these approaches on benchmark datasets for three laboratory processes. The measured data for the case studies are publicly available and are used for development, testing and benchmarking of system identification algorithms for nonlinear dynamic systems. Our experimental results show that SUHICLUST produces smaller models than LOLIMOT for two of the three datasets. In terms of error, ensembles of LOLIMOT models improve the predictive performance over that of a single LOLIMOT or SUHICLUST model.

Keywords: Fuzzy model identification, tree partitioning, fuzzy clustering, ensembles

1. INTRODUCTION

This paper considers the task of discrete-time modeling of nonlinear dynamic systems from input-output data. Using the *external dynamics approach* (Nelles, 2001), the task of empirical modeling of a dynamic system can be formulated as a regression problem of finding a difference equation that fits an observed behavior of the system.

One possible approach to the regression (i.e., nonlinear function approximation) problem is the *multimodel approach* (Murray-Smith and Johansen, 1997). Its idea is to combine several simple submodels and use them to describe the global behavior of the dynamic system. The operating region of the dynamic system is split into several subregions (partitions) and a simple local model is built for each subregion. The local models are combined by smooth interpolation into a complete global model.

Two fuzzy identification methods are considered, which are based on the multimodel approach. These methods determine the subregions by utilizing an automatic procedure which uses input-output data. In the literature, the problem of automatic defining of operating regions is solved using grid-based (Jang et al., 1997), fuzzy clustering (Jang et al., 1997), or tree partitioning methods (Nelles, 2001; Jang, 1994). The fuzzy methods evaluated here follow the last two principles.

Also, a local model is built for each identified region. Common approaches to parameter estimation in the local models are local or global optimization, frequently used with the least-squares cost function (Johansen and Babuška, 2003). The first estimates the parameters in each local model separately, while the second estimates the parameters in all local models jointly. The local optimization, which is faster and can handle noisy data, is employed in the fuzzy methods considered here.

This paper compares neuro-fuzzy identification methods, based on, or derived from the Lolimot method. In particular, it is concerned with building Takagi-Sugeno (TS) models (Takagi and Sugeno, 1985), by using LOLIMOT (Nelles, 2001) and SUHICLUST (Hartmann et al., 2013). Additionally, it considers the more accurate ensembles of fuzzy models (Aleksovski et al., 2015). The ensembles are built by using the bagging procedure and use LOLIMOT fuzzy models as base models. However, this paper does not include a comparison to adaptive or evolving methods because they are not of the same sort. Adaptive and evolving methods change their parameters with time and in the case of evolving methods also structure, while the models evaluated here are invariant to changes.

The LOLIMOT method (Nelles, 2001) uses an incremental construction scheme, and builds TS fuzzy models. It uses a tree partitioning method for defining the operating regions, i.e. for model structure identification. For the local models it uses local least squares estimation. The method is fast and noise-tolerant, due to the use of local estimation. The tree partitioning allows it to handle large dimensional problems efficiently: partitions have different sizes and are located in regions where the data allow for a finer partitioning. However, the partitioning is axis-

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.085

parallel: the partitions are hyperrectangles whose sides are parallel with the axes of the input space.

The SUHICLUST method was introduced by Hartmann et al. (2013). It unifies the strengths of supervised, incremental tree construction scheme of LOLIMOT with the advantages of product space clustering. By merging these two concepts, a fuzzy identification algorithm is obtained that, in contrast to LOLIMOT, enables axes-oblique partitioning and has highly flexible validity functions. According to Hartmann et al. (2011), the method produces fuzzy models with the same accuracy as LOLIMOT, but with fewer local models. The reproducibility of results is the same as with LOLIMOT and therefore better than with product space clustering. In Teslič et al. (2011) the SUHICLUST method was successfully used to model the drug absorption spectra process. Recently, an approach for the design of experiments based on the SUHICLUST method was proposed by Škrjanc (2015).

Ensembles (Krogh and Vedelsby, 1995; Dietterich, 2002), or committees of predictors, work by creating several *base models*. Each base model is an imperfect predictive model capturing a potentially different aspect of the system being modeled. Combining the imperfect predictions obtained from each base model should improve the predictive power over a single model and thus increase the accuraccy.

Ensembles based on the bootstrap resampling principle (Breiman, 1996) identify each base model from a modified version of the training data. First, several bootstrap samples are created from the training data, after which a model is built for each sample. The same principle is used by Aleksovski et al. (2015) for building ensembles of LOLIMOT models both for single-output and multi-output nonlinear dynamical systems.

The remainder of the paper is organized as follows. The next section describes the methodology: the use of LOLIMOT to build (single) fuzzy models and ensembles, and the use of SUHICLUST. Section 3 presents the experimental setup and describes the case studies, while Section 4 presents the results of the experimental evaluation. Finally, Section 5 concludes and outlines some directions for further work.

2. METHODOLOGY

This section introduces the fuzzy methods LOLIMOT and SUHICLUST, and describes the procedure for building ensembles of fuzzy models by using LOLIMOT. The fuzzy models evaluated in this paper have the Takagi- Sugeno (TS) form:

$$\hat{\boldsymbol{y}}(\boldsymbol{x}) = \sum_{j=1}^{m} \Phi_j(\boldsymbol{x}) f_{LMj}(\boldsymbol{x}). \quad (1)$$

Similarities between LOLIMOT and SUHICLUST. The methods evaluated in this paper use two different (automatic) procedures for defining operating regions (partitions), based on identification data. LOLIMOT uses tree partitioning, while SUHICLUST uses fuzzy clustering. The tree partitioning used in LOLIMOT is a computationally efficient approach which can handle large dimensional problems: it is able to create partitions of different sizes, which are smaller in those parts of the input space where finer partitioning is needed. Simpler grid partitioning methods suffer from the curse of dimensionality, as they create a complete grid over the input space. The second method evaluated here, SUHICLUST, defines the operating regions using a combination of fuzzy clustering, with the Gustafson-Kessel algorithm, and supervised learning (Hartmann et al., 2011).

2.1 LOLIMOT

The local linear model trees (LOLIMOT) method (Nelles, 2001) operates iteratively, using a tree partitioning procedure. In each iteration, it selects the worst performing partition, and splits it further. Local models are estimated at the end of each iteration for the new partitions.

In particular, each partition obtains a multi-dimensional Gaussian fuzzy membership function $\Phi_j(\boldsymbol{x})$, which is used to estimate a local affine model f_{LM_j}. Finally, the resulting model is a TS model where the output is calculated as:

$$\hat{y}(\boldsymbol{x}) = \sum_{j=1}^{m} (b_{j,0} + b_{j,1}x_1 + b_{j,2}x_2 + ... + b_{j,p}x_p)\Phi_j(\boldsymbol{x}), \quad (2)$$

and m is the number of local models.

Selection heuristic. In each iteration, LOLIMOT considers only one partition for further expansion. It selects the worst performing one, using the criterion of largest sum of squared errors. As LOLIMOT was designed for dynamic system identification, it is able to use simulation to evaluate the intermediate model in each step. It is performed using all available training data, and no averaging is used on the individual squared errors. [1]

After determining the worst performing partition, LOLIMOT considers splitting the partition in half, in every possible dimension. Each of these alternatives is evaluated by using a heuristic greedy evaluation function: (a) the partition boundaries are adjusted, and a fuzzy model with one more LM is created, (b) two new local models are estimated, and (c) simulation is used to evaluate the fit od the complete model to the training data.

Estimation of local model parameters. In each iteration of the method, the parameters of the newly added local models are estimated. The estimation begins by calculating the fuzzy membership function values. As a next step, these values are used in the weighted least square regression performed to obtain the parameters of the local models.

The membership functions determine the (fuzzy) membership of each data point to each of the partitions and the corresponding local models. LOLIMOT uses the multi-dimensional Gaussian membership function, whose center $\mathbf{c}$ is determined by the center of the partition, and standard deviation vector $\boldsymbol{\sigma}$ is calculated as 1/3 of the size of the partition (Nelles, 2001). Thus, the membership of a data point $\mathbf{x}$ to the j-th partition is calculated as

$$\mu_j(\mathbf{x}) = exp(-\frac{1}{2}\sum_{i=1}^{n}(\frac{x_i - c_i}{\sigma_i})^2). \quad (3)$$

[1] It is interesting to note that such a selection heuristic produces more partitions in the regions which contain more training data.

After the membership values for a data point are calculated, they are normalized across all partitions obtaining the validity function values $\Phi_j(\mathbf{x})$:

$$\Phi_j(\mathbf{x}) = \frac{\mu_j(\mathbf{x})}{\sum_{k=1}^{m} \mu_k(\mathbf{x})} \ . \tag{4}$$

The parameter estimation determines the coefficients $b_{j,u}$ of the local models (cf Eq. 2). LOLIMOT uses *local estimation* which estimates the parameters of each local model in isolation. As compared to the alternative *global estimation*, it has several advantages: it is faster, more stable, with better performance in noisy situations, and allows for interpretability of consequents (Hartmann et al., 2013).

For a local model j, the regression matrix $\boldsymbol{X}_j$ contains the values of the regressor variables:

$$\boldsymbol{X}_j = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}, \tag{5}$$

while the weighting matrix Q_j, contains the validity function values (Eq. 4):

$$Q_j = \begin{bmatrix} \Phi_j(\mathbf{x}_1) & 0 & \cdots & 0 \\ 0 & \Phi_j(\mathbf{x}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Phi_j(\mathbf{x}_n) \end{bmatrix}. \tag{6}$$

Defining the vector of outputs as $\boldsymbol{y} = [y_1, y_2, \cdots y_n]^T$, the parameter estimates for the local model j can be calculated using the well-known weighted least squares estimation formula

$$\hat{\boldsymbol{b}}_j = (\boldsymbol{X}_j^T Q_j \boldsymbol{X}_j)^{-1} \boldsymbol{X}_j^T Q_j \boldsymbol{y}. \tag{7}$$

2.2 Ensemble creation

The ensembles evaluated here are built by using the bagging procedure. The procedure starts by creating t bootstrap replicates, i.e., random samples with replacement, of the training data set D. The replicates have an equal number of data points as the training set. Each of the t replicates D_i is used to build one fuzzy model.

Thus, a collection of LOLIMOT models is built: $f_1, f_2, .., f_t$. Denoting the predictions of the t models of the ensemble with $\hat{f}_i(\mathbf{x})$, the overall prediction from the LOLIMOT ensemble is the average of the base model predictions:

$$\hat{y}(\mathbf{x}) = \frac{1}{t} \sum_{i=1}^{t} \hat{f}_i(\mathbf{x}). \tag{8}$$

It is worth noting that the bootstrap samples cannot be used for performing simulation during the building of a LOLIMOT model, due to the requirement of having time contiguous data points. Hence, the complete training set D is also made available when building each base model (Aleksovski et al., 2015).

2.3 SUHICLUST

The supervised hierarchical clustering (SUHICLUST) method differs from LOLIMOT in the problem space partitioning algorithm. Instead of iteratively checking each dimension for splitting the worst performing fuzzy region (local partition), SUHICLUST performs the split based on the main eigenvector. The fuzzy regions are then optimized using the Gustafson-Kessel (GK) algorithm. As in LOLIMOT, SUHICLUST selects the worst performing fuzzy region (cluster) based on the sum of squared errors. The local regions in SUHICLUST are defined by the fuzzy covariance matrix and by the mean of data in that region. The local regions are called clusters. The mean of the data in the local region represent a cluster center. The fuzzy covariance matrices and clusters' centers are obtained from the GK clustering algorithm.

When a worst performing local model is found, the cluster on which this model is valid is split. The cluster's fuzzy covariance matrix is decomposed by using singular value decomposition. With this, the eigenvectors of data around the cluster center are obtained. The main direction of the data expansion is the direction with the largest eigenvalue. Therefore the main eigenvector is used to split the cluster. Two new clusters are created, one by adding the main eigenvector to the cluster center and the other by subtracting the main eigenvector from the cluster center.

Next, GK clustering is performed by using the two newly created clusters and the local data. The local data are data samples that have validity function values for the split cluster higher than a user defined threshold (usually 0.5). Global optimization of local regions is performed by using all generated clusters and all data. This is again done by GK clustering. At the end of each iteration, the local models' parameters are identified by using Eq. 7.

The SUHICLUST method has four important parameters. The first two are the same as with LOLIMOT. One is the error threshold and the other is the maximal number of clusters. These two parameters are used as stopping criteria for the identification. Next is the validity threshold that defines the local data and the fuzziness parameter η_m. This parameter is used to control the smoothness of the membership functions (cf Eq. 3). Namely, the standard deviation σ_i in Eq. 3 is multiplied by a factor of η_m in SUHICLUST. The standard deviations σ are determined by SUHICLUST as a square roots of diagonal elements of fuzzy covariance matrices.

Further, we have also two parameters connected to the GK clustering. One is the fuzziness factor (usually set between 1 and 2) and the stopping threshold for clustering (usually set to 0.01). When one uses GK clustering with improved variance estimation, two additional parameters have to be set as explained by Babuška et al. (2002).

3. EXPERIMENTAL EVALUATION SETUP

This section describes the setup for the evaluation of LOLIMOT, bagging with LOLIMOT and SUHICLUST for dynamic system identification. It starts by describing the experimental setup (i.e. the procedure of evaluation and evaluation measure). This is followed by descriptions of the three case studies. Finally, it reports the datasets, regressors and parameters of the methods used.

3.1 Experimental setup

The experimental setup uses input-output data for each of the case studies. The data are split into two sets: a training set and a testing set. The model is built by using the training set and evaluated by using the testing set. The error reported is in terms of normalized root mean squared error (NRMSE), also known as root relative mean-squared error (RRMSE):

$$e^2_{NRMSE} = \frac{\sqrt{\sum (y_i - \hat{y}_i)^2}}{\sqrt{\sum (y_i - \bar{y})^2}}, \tag{9}$$

where $\bar{y}$ denotes the average of the output variable.

The models are evaluated in terms of output error, by using simulation: The predicted values of the model at time point k is used in the regressor vector at time point $k+1$. This presents a more control engineering oriented and stringent type of evaluation of the model, given the realistic possibility of error accumulation.

3.2 Case study: coupled electric drives

The coupled electric drives system (Wellstead, 1979; Wigren, 1990) consists of two electric motors which are used to drive a pulley by using a flexible belt (Fig 1 (a)). The pulley is held by a spring, resulting in a lightly damped dynamic mode. The electric drives can be individually controlled allowing the tension and the speed of the belt to be simultaneously controlled. The drive control is symmetric around zero, hence both clockwise and counter clockwise movement is possible. Here the focus is only on the speed control system. The reason is that the angular speed of the pulley is measured with a pulse counter and this sensor is insensitive to the sign of the velocity. Following the sensor, analogue low pass filtering and anti-aliasing filtering are applied. The dynamic effects are generated by the electric drive time constants, the spring and the analogue low pass filtering. The latter has a quite limited effect on the output and may be neglected.

A discrete time Wiener model of the system (Wigren and Schoukens, 2013) is given by:

$$y(t) = \frac{b_1 q^{-1} + b_2 q^{-2} + b_3 q^{-3}}{1 + f_1 q^{-1} + f_2 q^{-2} + f_3 q^{-3}} u(t) \tag{10}$$

$$y_n(t) = |y(t)|, \tag{11}$$

where b_i, f_i; $i = 1, .., 3$ are parameters, $u(t)$ is the input signal, $y_n(t)$ is the output signal. The input signal $u(t)$ used was a PRBS with a clock period of 5 times the sampling period. The signal was switching between $-1\ V$ and $+3\ V$, resulting in the process changing the belt rotation direction frequently. The input-output data was recorded with a sampling period of 20 ms. The data for this case study are shown in Fig. 1 (b).

3.3 Case study: two cascaded tanks

This process is a fluid level control system consisting of two cascaded tanks with free outlets fed by a pump (Wigren, 2006). The water is transported by the pump to the upper

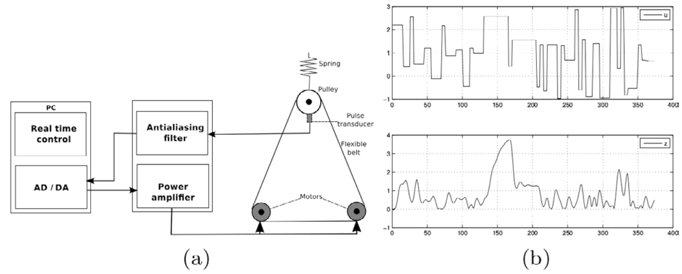

Fig. 1. (a) A schematic diagram of the coupled electric drives. (b) The training data used in this study.

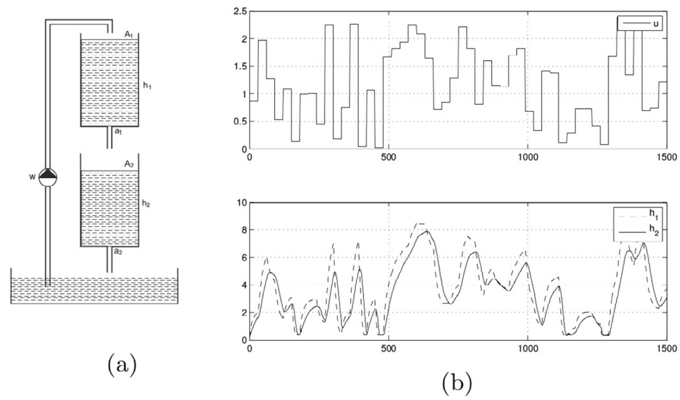

Fig. 2. (a) A schematic diagram of the two cascaded tanks laboratory process. (b) The training data used in this study.

of the two tanks. The process is depicted in Fig. 2 (a). The input signal to the process is the voltage applied to the pump $u(t)$ and the two output signals consist of measurements of the water levels of the tanks h_1 and h_2. Since the outlets are open, and since the tanks are deep with large vertical extension, the result is a significantly non-linear dynamics that varies with the level of water.

The equations governing the system are as follows (Wigren and Schoukens, 2013):

$$\begin{bmatrix} \frac{dh_1}{dt} \\ \frac{dh_2}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{a_1\sqrt{2g}}{A_1}\sqrt{h_1} + \frac{1}{A_1} k u(t) \\ -\frac{a_2\sqrt{2g}}{A_2}\sqrt{h_2} + \frac{a_1\sqrt{2g}}{A_2}\sqrt{h_1} \end{bmatrix}, \tag{12}$$

where A_1, A_2, a_1, a_2 denote the areas of the two tanks and the two effluent areas, respectively. The voltage to input flow conversion constant is k, while the applied voltage to the pump is $u(t)$. The data available for this case study are shown in Fig. 2 (b).

3.4 Case study: silver box

This case study concerns an electronic nonlinear feedback laboratory system (denoted ”the silver box”) (Pintelon and Schoukens, 2012). Analogue electrical circuitry is used to generate data representing a nonlinear mechanical resonating system with a moving mass m, a viscous damping d, and a nonlinear spring $k(y)$. The purpose of the electrical circuit is to relate the displacement $y(t)$ to the force $u(t)$ according to the following differential equation:

$$m\frac{d^2y(t)}{dt} + d\frac{dy(t)}{dt} + k(y(t))y(t) = u(t)\ . \tag{13}$$

The nonlinear spring is described by using a static position-dependent stiffness

$$k(y(t)) = a + by^2(t) \ . \tag{14}$$

We have used 10 000 data points for training and 4000 for testing, which are shown in Fig. 3.

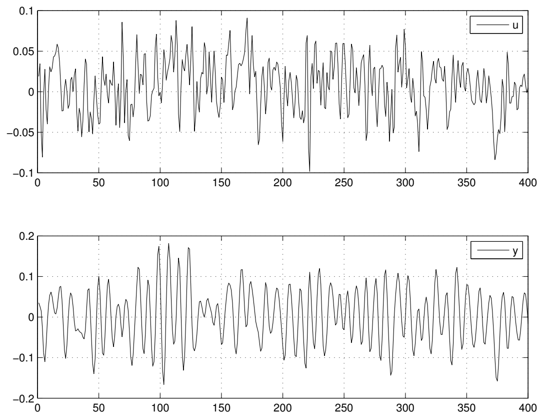

Fig. 3. The first 400 training data points, used for the silver box case study.

3.5 Datasets and Regressors

Using measured data for the three case studies, we defined three datasets: coupled electric drives (CED), two cascaded tanks (TCT), and silver box (SB). Using a trial-and-error procedure, we selected regressors for the three datasets. The regressors are reported in Table 1, where it is possible to see that the maximal lag used was 9, 4, and 4 for the three datasets respectively. The same regressors were then used for all methods that we evaluated.

Table 1. The datasets and the regressors used

Dataset	**#Data points (training / test)**	**Output var.**	**Regressors selected**
CED	374 / 126	$z(k)$	$u(k-6)$, $u(k-4)$, $u(k-1)$, $z(k-9)$, $z(k-8)$, $z(k-7)$, $z(k-6)$, $z(k-4)$, $z(k-1)$
TCT	1500 / 1000	$h_2(k)$	$u(k-4)$, $u(k-2)$, $h_1(k-4)$, $h_2(k-4)$, $h_2(k-1)$
SB	10000 / 4000	$y(k)$	$u(k-3)$, $u(k-2)$, $u(k-1)$, $y(k-4)$, $y(k-2)$, $y(k-1)$

3.6 Parameters of the methods

The parameter settings of the methods are described next. The parameters of LOLIMOT are discussed first, followed by a discussion of the parameters of the ensemble method and the parameters of SUHICLUST.

Parameters of LOLIMOT and bagging. The single parameter that is set for the model tree algorithm is the number of local models. It was determined according to the criterion of minimal NRMSE error of simulation, on the training set. The values thus selected were 30 local models for all three datasets, which was also used for bagging. We vary the number of base models in the ensemble, considering up to 64 LOLIMOT models. The results for all of these are shown in graphical form in Figure 4. Based on these results, we examine the minimal number of models in the ensemble.

Parameters of SUHICLUST. The number of local models used in SUHICLUST was also according to the criterion of minimal NRMSE error of simulation, on the training set. Maximal number of local models was set to 20. The values obtained were 11, 18 and 20 for the three datasets CED, TCT and SB respectively. In all three cases the validity threshold was set to 0.5, the fuzziness factor η_m to 0.5 and criterion for stopping the GK clustering to 0.1. The fuzziness factor for GK clustering was for CED experiment set to 1.3, for TCT experiment to 1.1 and for SB experiment to 1.2.

4. RESULTS

This section presents the results of the performance evaluation of LOLIMOT, bagging with LOLIMOT and SUHICLUST. It first reports the results in terms of the number of local models, and then in terms of squared error.

Number of local models. The experimental results shown here report the number of local models of LOLIMOT and SUHICLUST. In more detail, Table 2 gives the number of local models (#LMs) needed in order to achieve a specific modeling goal of an error lower than e^2_{NRMSE} on the test set. It can be seen that for CED and TCT SUHICLUST requires less LMs then LOLIMOT, while showing error of similar magnitude. The number of LMs required differs by 9 and 6 LMs respectively. However, for SB, LOLIMOT satisfies the modeling goal with 5 LMs less.

Table 2. Comparison of LOLIMOT and SUHICLUST

Dataset	**Modeling goal** (e^2)	**LOLIMOT** (e^2 / #LMs)	**SUHICLUST** (e^2 / #LMs)
CED	0.1000	0.0907 / 21	0.0997 / 12
TCT	0.0350	0.0342 / 19	0.0333 / 13
SB	0.0700	0.0628 / 6	0.0660 / 11

Squared error. The results in terms of NRMSE error on the test sets are presented in Figure 4 presents. Note that there are 5 dashed lines for each size of the ensemble, showing the performance of 5 runs of bagging with different random seeds. This was included in order to illustrate the effect of randomness as used in the procedure of bagging.

We can conclude that bagging with LOLIMOT improves the error over a single LOLIMOT model. The improvement is visible for all three datasets. An exception holds only for CED, for which one run of bagging does not clearly improve over the error of a single LOLIMOT model (the other 4 runs do improve).

Regarding the number of base models in the ensemble, the results for TCT and SB show that a low number of base models (as few as 10) in the ensemble is sufficient for improving the performance over LOLIMOT. For the CED dataset, however, 40 base models are needed to stabilize the variance of the NRMSE error and improve over LOLIMOT's performance. The comparison of ensembles of LOLIMOT models to SUHICLUST shows that ensembles can achieve similar (with 10 models) or better performance (with 40 models) than a SUHICLUST model.

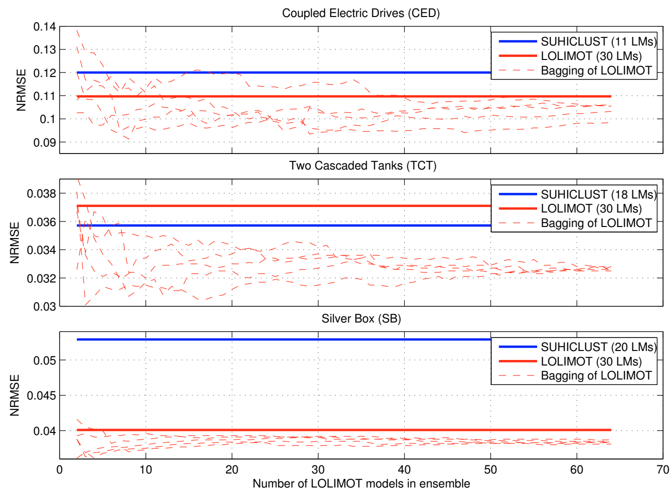

Fig. 4. NRMSE results for the three case studies.

5. CONCLUSIONS AND FURTHER WORK

This paper evaluated LOLIMOT, SUHICLUST and ensembles of LOLIMOT models, built by using measured input-output data. The evaluation was performed on three benchmark case studies of modeling nonlinear dynamic systems. The case studies cover nonlinear laboratory processes: coupled electric drives, cascaded tanks and silver box.

In terms of size of the fuzzy models, for two datasets SUHICLUST produced smaller models as compared to LOLIMOT, with similar error. For the remaining dataset, the model built by LOLIMOT was smaller. In terms of error, ensembles of LOLIMOT models performed better than a single LOLIMOT or SUHICLUST model. As further work, we would like to evaluate different ensemble approaches to learning fuzzy models, such as boosting.

ACKNOWLEDGEMENTS

This work was supported by the Slovenian Research Agency (through the grants P2-0001 and P2-0103) and the European Commission (through the grant MAESTRA, FP7-ICT-612944). Thanks to Oliver Nelles for providing the source code of LOLIMOT.

REFERENCES

Aleksovski, D., Kocijan, J., and Džeroski, S. (2015). Ensembles of fuzzy linear model trees for the identification of multi-output systems. IEEE Transactions on Fuzzy Systems.

Babuška, R., van der Veen, P., and Kaymak, U. (2002). Improved covariance estimation for gustafson-kessel clustering. In *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, 1081 – 1085. IEEE.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140.

Dietterich, T.G. (2002). Ensemble learning. In *The Handbook of Brain Theory and Neural Networks*, 405–408. The MIT Press, 2nd edition.

Hartmann, B., Nelles, O., Škrjanc, I., and Sodja, A. (2013). Supervised hierarchical clustering (SUHICLUST) for nonlinear system identication. In *IEEE Symposium on Computational Intelligence in Control and Automation 2009*, 41 – 48. IEEE.

Hartmann, B., Bänfer, O., Nelles, O., Sodja, A., Teslič, L., and Škrjanc, I. (2011). Supervised hierarchical clustering in fuzzy model identication. *IEEE Transactions on Fuzzy Systems*, 19(6), 1163–1176.

Jang, J.S.R. (1994). Structure determination in fuzzy modeling: A fuzzy CART approach. In *Proceedings of the Third IEEE Conference on Fuzzy Systems*, 480–485. IEEE.

Jang, J.S.R., Sun, C.T., and Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing–A Computational Approach to Learning and Machine Intelligence*. Prentice Hall.

Johansen, T.A. and Babuška, R. (2003). Multiobjective identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, 11(6), 847–860.

Krogh, A. and Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In *Proceedings of the 8th International Conference on Advances in Neural Information Processing Systems*, 231–238. MIT Press.

Murray-Smith, R. and Johansen, T. (eds.) (1997). *Multiple Model Approaches to Modelling and Control*. Taylor & Francis.

Nelles, O. (2001). *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer.

Pintelon, R. and Schoukens, J. (2012). *System Identification: A Frequency Domain Approach*. John Wiley & Sons.

Škrjanc, I. (2015). Evolving fuzzy-model-based design of experiments with supervised hierarchical clustering. *IEEE Transactions on Fuzzy Systems*, 23(4), 861–871.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1), 116–132.

Teslič, L., Hartmann, B., Nelles, O., and Škrjanc, I. (2011). Nonlinear system identication by Gustafson-Kessel fuzzy clustering and supervised local model network learning for the drug absorption spectra process. *IEEE Transactions on Neural Networks*, 22(12), 1163–1176.

Wellstead, P.E. (1979). *Introduction to Physical System Modelling*. Academic Press.

Wigren, T. (1990). *Recursive Identification Based on the Nonlinear Wiener Model*. Ph.D. thesis, Uppsala University, Uppsala, Sweden.

Wigren, T. (2006). Recursive prediction error identification and scaling of non-linear state space models using a restricted black box parameterization. *Automatica*, 42(1), 159–168.

Wigren, T. and Schoukens, J. (2013). Three free data sets for development and benchmarking in nonlinear system identification. In *European Control Conference 2013*, 2933–2938. IEEE.