# Autoregressive Time Series Prediction by Means of Fuzzy Inference Systems Using Nonparametric Residual Variance Estimation

Federico Montesino Pouzols *, Amaury Lendasse

*Department of Information and Computer Science, Helsinki University of Technology*

*P.O. Box 5400 FI-02015 HUT Espoo, Finland*

Angel Barriga Barros

[b]*Department of Electronics and Electromagnetism*

*University of Seville*

*Avda. Reina Mercedes s/n, E-41012 Seville, Spain.*

---

* Corresponding author. Phone: +358-9-451-5237. FAX: +358-9-451-3277.
*Email addresses:* fedemp@cis.hut.fi (Federico Montesino Pouzols), lendasse@hut.fi (Amaury Lendasse), barriga@us.es (Angel Barriga Barros).

**Abstract**

We propose an automatic methodology framework for short- and long-term prediction of time series by means of fuzzy inference systems. In this methodology, fuzzy techniques and statistical techniques for nonparametric residual variance estimation are combined in order to build autoregressive predictive models implemented as fuzzy inference systems. Nonparametric residual variance estimation plays a key role in driving the identification and learning procedures. Concrete criteria and procedures within the proposed methodology framework are applied to a number of time series prediction problems. The learn from examples method introduced by Wang and Mendel (W&M) is used for identification. The Levenberg-Marquardt (L-M) optimization method is then applied for tuning. The W&M method produces compact and potentially accurate inference systems when applied after a proper variable selection stage. The L-M method yields the best compromise between accuracy and interpretability of results, among a set of alternatives. Delta test based residual variance estimations are used in order to select the best subset of inputs to the fuzzy inference systems as well as the number of linguistic labels for the inputs. Experiments on a diverse set of time series prediction benchmarks are compared against least-squares support vector machines (LS-SVM), optimally-pruned extreme learning machine (OP-ELM), and $k$-NN based autoregressors. The advantages of the proposed methodology are shown in terms of linguistic interpretability, generalization capability and computational cost. Furthermore, fuzzy models are shown to be consistently more accurate for prediction in the case of time series coming from real-world applications

**Key words**

Fuzzy inference systems, Time series prediction, Nonparametric regression, Supervised learning, Nonparametric residual variance estimation

## 1 Introduction

Time series prediction and analysis in general is a recurrent problem in virtually all areas of natural and social sciences as well as in engineering. In the time series prediction field, prediction accuracy is not the only major goal. Understanding the behavior of time series and gaining insight into their underlying dynamics is a highly desired capability of time series prediction methods [52].

In the past, conventional statistical techniques such as AR and ARMA models have been extensively used for forecasting [6]. However, these techniques have limited capabilities for modeling time series data, and more advanced nonlinear methods including artificial neural networks have been frequently applied with success [9].

Fuzzy logic based modeling techniques are appealing because of their interpretability and potential to address a broad spectrum of problems. In particular, fuzzy inference systems exhibit a combined description and prediction capability as a consequence of their rule-based structure [50]. The application of fuzzy inference systems to time series modeling and prediction dates back to [51], in which the authors develop the well known learn from examples identification algorithm for fuzzy inference systems and use the Mackey-Glass time series as a validation case. Nevertheless, despite its good performance in terms of accuracy and interpretability, fuzzy inference systems have seen little application in the field of time series prediction as compared to other nonlinear modeling techniques such as artificial neural networks and support vector machines.

The methodology proposed in this paper is intended to apply to crisp time series, i.e. those time series consisting of crisp values, as opposed to other kinds of values, such as interval and fuzzy values. That is, we propose here an automatic methodology framework to perform autoregressive prediction of crisp time series by means of fuzzy

inference systems using nonparametric residual variance estimation [37]. We will call fuzzy autoregressors those autoregressors implemented as fuzzy inference systems. This is not to be confused with what is usually called fuzzy regression in the literature [8].

When developing fuzzy inference systems for time series prediction, many questions remain still open: how many and what inputs to the inference system must be defined? To what extent the theoretical universal approximation capability of fuzzy systems is achieved with existing techniques? What are the best fuzzy techniques for these tasks? How to perform long-term prediction?

In practice, one finds two problems when building a fuzzy model for a time series: choosing variables or inputs to the inference system, and identifying the structure of the system (linguistic labels and rule base). Once these steps have been accomplished, the fuzzy model can be tuned through supervised learning techniques. We propose an automatic methodology framework to address these two problems using fuzzy techniques and nonparametric residual variance estimation techniques in an intertwined manner.

The first problem can be addressed by means of a priori feature selection techniques based on nonparametric residual variance estimation, which also provide an estimate of the error of the most accurate nonlinear model that can be built without overfitting. The second problem is addressed by data-driven techniques for identification of fuzzy systems from numerical examples [16], such as the algorithm by Wang and Mendel (W&M) [50,51] and the fuzzy identification algorithm based on clustering by Chiu [12,36], two well established methods among the many alternatives proposed throughout the years [23,19,43,34,2,26,44].

This paper also addresses a recent challenge in the field of time series prediction: long-term prediction (as a generalization to short-term prediction), for which lack of

information and accumulated errors pose additional difficulties. Furthermore, time series coming from real world applications, in addition to series generated either numerically or under controlled laboratory conditions are analyzed.

With these premises we propose a methodology for building simple, interpretable yet highly accurate fuzzy inference models. These will be compared against least-squares support vector machines (LS-SVM) [48], a well established method in the field of time series prediction, that has been shown to be highly accurate. For further comparison, we will also show the results obtained using optimally-pruned extreme learning machine (OP-ELM) and $k$-nearest neighbors ($k$-NN) models.

The article is organized as follows. The next section outlines a nonparametric residual variance estimation method that will be used for both variable and proper model complexity selection. In section 3 we propose a methodology framework and one concrete implementation that uses well known algorithms for identifying and optimizing fuzzy inference systems. Section 4 illustrates the methodology through a case study. Finally, sections 5 and 6 present and further discuss experimental results for a number of time series benchmarks from diverse fields of application. In appendices A and B we provide further comparisons of different modeling techniques.

## 2 Nonparametric Residual Variance Estimation: Delta Test

Nonparametric residual variance estimation (or nonparametric noise estimation, NNE) is a well-known technique in statistics and machine learning, finding many applications in nonlinear modeling [24]. NNE methods can be applied to recurrent problems such as variable and model structure selection. These methods are not however in widespread use in the machine learning community as most work has been done to date within the statistics community.

Delta Test (DT), introduced for time series in 1994 [41], is a NNE method, i.e., it estimates the lowest mean square error (MSE) that can be achieved by a model without overfitting the training set [24]. Given $N$ multiple input-single output pairs, $(\bar{x}_i, y_i) \in \mathbb{R}^M \times \mathbb{R}$, the theory behind the DT method considers that the mapping between $\bar{x}_i$ and $y_i$ is given by the following expression:

$$y_i = f(\bar{x}_i) + r_i,$$

where $f$ is an unknown perfect fitting model and $r_i$ is the noise. DT is based on hypothesis coming from the continuity of the regression function. When two inputs $x$ and $x'$ are close, the continuity of the regression function implies that the corresponding outputs, $f(x)$ and $f(x')$ will be close enough. When this implication does not hold, it is due to the influence of the noise.

Let us denote the first nearest neighbor of the point $\bar{x}_i$ in the set $\{\bar{x}_1, \ldots, \bar{x}_N\}$ by $\bar{x}_{NN}$. Then the DT, $\delta$, is defined as follows:

$$\delta = \frac{1}{2N} \sum_{i=1}^{N} \left| y_{NN(i)} - y_i \right|^2,$$

where $y_{NN(i)}$ is the output corresponding to $\bar{x}_{NN(i)}$. For a proof of convergence, refer to [28,29]. DT is an unbiased and asymptotically perfect estimator with a relatively fast convergence [29] and is useful for evaluating nonlinear correlations between two random variables, namely, input-output pairs. DT can be seen as part of a more general NNE framework known as the Gamma Test [24]. Despite the simplicity of DT, it has been shown to be a robust method in real world applications [28]. This method will be used in the next sections for a priori input selection.

## 3 Methodology Framework for Time Series Prediction with Fuzzy Inference Systems

Consider a discrete time series as a vector, $\bar{y} = y_1, y_2, \ldots, y_{t-1}, y_t$, that represents an ordered set of values, where $t$ is the number of values in the series. The problem of predicting one future value, $y_{t+1}$, using an autoregressive model (autoregressor) with no exogenous inputs can be stated as follows:

$$\hat{y}_{t+1} = f_r(y_t, y_{t-1}, \ldots, y_{t-M+1}),$$

where $\hat{y}_{t+1}$ is the prediction of model $f_r$ and $M$ is the number of inputs to the regressor, i.e., the regressor size.

Predicting the first unknown value requires building a model, $f_r$, that maps regressor inputs (known values) into regressor outputs (predictions). When a prediction horizon higher than 1 is considered, the unknown values can be predicted following two main strategies: recursive and direct prediction.

The recursive strategy applies the same model recursively, using predictions as known data to predict the next unknown values. For instance, the third unknown value is predicted as follows:

$$\hat{y}_{t+3} = f_r(\hat{y}_{t+2}, \hat{y}_{t+1}, y_t, y_{t-1}, \ldots, y_{t-M+3}).$$

Recursive prediction is the most simple and intuitive strategy and does not require any additional modeling after an autoregressor for 1 step ahead prediction is built. However, recursive prediction suffers from accumulation of errors. The longer the prediction horizon is, the more predictions are used as inputs. In particular, for prediction horizons greater than the regressor size, all inputs to the model are predictions.

Direct prediction requires that the process of building an autoregressor be applied for

each unknown future value. Thus, for a maximum prediction horizon $H$, $H$ direct models are built, one for each prediction horizon $h$:

$$\hat{y}_{t+h} = f_h(y_t, y_{t-1}, \ldots, y_{t-M+1}), \text{ with } 1 \leq h \leq H.$$

While building a prediction system through direct prediction is more computationally intensive (as many times as values are to be predicted) it is also straightforward to parallelize. As opposed to recursive prediction, direct prediction does not suffer from accumulation of prediction errors.

In this paper, we follow the direct prediction strategy. In order to build each autoregressor, a fuzzy inference system is defined as a mapping between a vector of crisp inputs and a crisp output. Let us rename the inputs $y_t, y_{t-1}, \ldots, y_{t-M+1}$ as $y_1, \ldots, y_M$ for simplicity. This way, assuming all ($M$) inputs are used, the fuzzy autoregressor for prediction horizon $h$ can be expressed as a set of $N_h$ fuzzy rules of the following form:

$$R_i^h : \text{ IF } y_1 \text{ is } L_1^{i,h} \text{ AND } y_2 \text{ is } L_2^{i,h} \text{ AND } \ldots \text{ AND } y_M \text{ is } L_M^{i,h} \text{ THEN } \hat{y}_{t+h} \leftarrow \mu_{R_i^h},$$

where $i = 1, \ldots, N_h$, and the fuzzy sets $L_j^{i,h} \in \{L_{j,k}^h\}, k = 1 \ldots, n_j^h, j = 1, \ldots, M$, with $n_j^h$ being the number of linguistic labels defined for the $j$th input variable. $L_j^{i,h}$ are the fuzzy sets representing the linguistic terms used for the $j$th input in the $i$th rule of the fuzzy model for prediction horizon $h$. $\mu_{R_i^h}$ are the consequents of the rules and can take different forms. For example, in a system with two inputs, if $L_1^{i,h}$ is renamed $LOW_1$ and $L_2^{i,h}$ is renamed $HIGH_2$, the $i$th rule for horizon 1, $R_i^1$, would have the following form:

$$\text{IF } y_t \text{ was } LOW_1 \text{ AND } y_{t-1} \text{ was } HIGH_2 \text{ THEN } \hat{y}_{t+h} \leftarrow \mu_{R_i^h}.$$

Depending on the fuzzy operators, inference model and type of membership functions (MFs) employed, the mapping between inputs and outputs can have different formulations. In principle, the methodology proposed in this paper can be applied for

any combination of types of MFs, operators and inference model, but the selection can have a significant impact on practical results.

As a concrete implementation for this paper, we use the minimum as T-norm for conjunction operations, Gaussian MFs for inputs, singleton outputs, and product inference of rules. Defuzzification is performed using the fuzzy mean method, i.e., zero-order Takagi-Sugeno systems are defined. Thus, the result of the inference process is a weighted average of the singleton consequents. This inference scheme was chosen in order to keep systems as simple and interpretable as possible.

Therefore, in this particular case a fuzzy autoregressor for prediction horizon $h$ can be formulated as follows:

$$\mathcal{F}_h(\bar{y}) = \frac{\sum_{i=1}^{N_h} \left( \mu_{R_i^h} \cdot \min_{1 \leq j \leq M} \mu_{L_j^{i,h}}(y_j) \right)}{\sum_{i=1}^{N_h} \min_{1 \leq j \leq M} \mu_{L_j^{i,h}}(y_j)}, \tag{1}$$

where $N_h$ is the number of rules in the rule base for horizon $h$, $\mu_{R_i^h}$ are singleton output values, and $\mu_{L_j^{i,h}}$ are Gaussian MFs for the inputs. Thus, each fuzzy set defined for the input linguistic terms, $L_{j,k}^h$ (for horizon $h$, and the $k$th term defined for the $j$th input), is characterized by an MF having the following form:

$$\mu_{L_{j,k}^h} = exp\left[-(y_j - c_{j,k,h})^2 / 2\sigma_{j,k,h}^2\right],\ k = 1 \ldots, n_j^h,\ j = 1, \ldots, M,\ h = 1, \ldots, H,$$

where $c_{j,k,h}$ are the centers of the MFs and $\sigma_{j,k}$ are the widths.

Together with the number of rules of a system, the total number of MFs can be seen as a measure of the complexity of a fuzzy inference system, or the structure of the equivalent artificial neural network. If the same $n_j^h$ number of linguistic terms is set for every input, then the total number of input MFs per prediction horizon is $M \cdot n_j^h$.

The problem of building a regressor can be precisely stated as that of defining a proper

number and configuration of MFs and building a fuzzy rule base from a data set comprising $t$ samples from a time series such that the fuzzy systems $\mathcal{F}_h(\bar{y})$ closely predict the $h$th next values of the time series. The error metric to be minimized is the mean squared error (MSE).

In this paper, we propose a methodology framework in which a fuzzy inference system is defined for each prediction horizon throughout the stages shown in figure 1. These stages and how they are specifically implemented in this paper are detailed in the following subsections.

### *3.1 Variable Selection*

In principle, the whole set of known past values of a time series may influence the unknown future values. However, using all known values as inputs to a time series autoregressor does not necessarily improve its accuracy. As the number of inputs increases, and the known data become more sparse in a high-dimensional space, building a model gets more and more complex. This is the well known "curse of dimensionality" problem [5].

In the time series prediction field, the application of variable selection methods has been shown to provide several advantages, such as reducing the model complexity and increasing the accuracy of predictions [46]. A proper choice of input variables can alleviate the curse of dimensionality problem while all the relevant data are still available for building a model. As first step in the methodology, DT estimates are employed so as to perform an a priori selection of the optimal subset of inputs from the initial set of $M$ inputs, given a maximum regressor size $M$.

Variable selection requires a selection criterion. We use the result of the DT applied to a

particular variable selection as a measure of the goodness of the selection. The input selection that minimizes the DT estimate, and thus the achievable MSE, is chosen for the next stages.

In addition, a selection procedure is required. For small (roughly up to around 10-20) regressor sizes, an exhaustive evaluation of DT for all the possible selections (a total of $2^M - 1$) is feasible. We will call this procedure *exhaustive DT search*. Its main advantage is that the optimal selection is guaranteed. However, its algorithmic order is exponential and it is thus unfeasible for high regressor sizes.

For higher regressor sizes, different search methods that partially scan the space of possible selections can be applied. In particular, forward-backward search of selections (FBS) [46] provides good results while being simple and efficient. This procedure combines both forward and backward selection. FBS can be started alternatively from random selections or selections for lower regressor sizes performed by means of exhaustive search. As a partial search procedure, FBS does not guarantee the optimality of the selection, however it provides a convenient balance between performance and computational requirements.

NNE based selection can be classified into the set of model independent methods for input selection. These methods select inputs a priori, i.e., the selection stage is based only on the dataset and does not require to build models. Thus, the computational cost of DT based selection is lower than that of the model dependent cases, in which input selection is addressed as a generalization error minimization problem, using leave-one-out, bootstrap or other resampling techniques [28].

### *3.2 System Identification and Tuning*

Usually, defining a fuzzy inference model from data requires two steps: the identification of the structure and the optimization of parameters [23,43]. The identification and tuning stage of our methodology comprises three substages, see figure 1, that are performed iteratively and in a coordinated manner. The whole process is driven by the third (complexity selection) substage, until a system that satisfies a training error condition derived from the DT estimate is constructed.

#### *3.2.1 Substage 2.1: System identification*

In this substage, the structure of the inference system (linguistic labels and rule base) is defined by means of an automatic fuzzy systems identification algorithm. The set of inputs is fixed after the previous (variable selection) stage. Regardless of the identification algorithm used, one or more parameters are usually required that specify the potential complexity of the inference system. Thus, the desired boundaries of complexity for the systems being built are additional inputs to the identification process.

The identification substage, as well as the next (tuning) substage are iteratively performed for increasing degrees of complexity. The concrete procedure used to explore different complexities depends on the identification and tuning algorithms applied.

For the concrete implementation analyzed in this paper, identification is performed using the W&M algorithm driven by the DT estimate. The W&M algorithm is based on the "learn by example" principle and considers a uniform grid partition of the universes of discourse of the inputs, which are proper characteristics for modeling time series in an interpretable manner. Though a number of modifications and derived algorithms have been proposed, for the sake of simplicity and interpretability we adhere to the original

specification of the algorithm for generating fuzzy inference rules directly from input-output data pairs [50] as implemented in version 3.2 of the Xfuzzy design environment [40].

In the case of the W&M algorithm, the number of MFs per input must be specified a priori. This can be done in an automated manner thanks to the use of the DT estimate in the iterative identification and tuning process. Our approach is to explore the set of possible systems starting from the lowest possible number of linguistic labels. This is an iterative process driven by the third substage (complexity selection), as explained below. For simplicity, the same number of linguistic labels is used for each input. In this case, the complexity boundaries would be thus specified as a maximum number of linguistic labels to explore.

### *3.2.2 Substage 2.2: System Tuning*

We consider a tuning step in the methodology as a substage separated from the identification substage. Note that in some cases (as for example in the algorithm by Higgins and Goodman [18]), these two substages can be integrated into a standalone algorithm. The tuning process is driven by one or more error metrics.

As concrete implementation for this paper we apply the Levenberg-Marquardt second order optimization method [3] for supervised learning, driven by the normalized MSE (NMSE) [1] . A number of supervised learning and optimization methods have been compared for this study, including gradient descent, probabilistic, second order, and conjugate gradient methods. The Levenberg-Marquardt method was selected on the basis of the following observations. First, it produces systems with the lowest number of MFs among the set of alternatives tested. Second, it yields systems almost as accurate as the

[1] Normalization is performed against the squared range of the series.

most accurate alternative. This is further discussed in appendix B, where several methods are compared.

All the parameters of the MFs of every input and output are adjusted so that the training error is minimized, i.e., self-tuning inference systems are defined. The learning algorithm applied is the Levenberg-Marquardt method as implemented in Xfuzzy [39].

### *3.2.3 Substage 2.3: Complexity Selection*

As last step in the process of identifying and tuning fuzzy autoregressors, the proper complexity of the estimated best autoregressor is selected depending on the DT estimate. The iterative identification and tuning stage stops when a system is built such that its training error is equal to or lower than the DT estimate or a threshold based on the DT estimate. Since identification and tuning iterations are performed for increasing complexities, the simplest system that satisfies the DT based error condition is selected.

For the particular implementation used in this paper, the complexity of fuzzy systems is measured as the number of linguistic labels per input. Thus, this substage selects the system with the lowest number of labels per input that has a training error equal to or lower than an optimal error threshold based on the DT estimate.

We note that the DT estimate is an estimate of the lowest possible error, i.e, the error that an optimal model would achieve. Since we cannot expect the models we will apply to be perfect, we introduce a DT based threshold. The DT based threshold, equal to or greater than the DT estimate, will be defined and validated experimentally in the next section.

Regarding the convergence and guarantee of finalization of this iterative process, neither the identification algorithm or the optimization method used here guarantee any error bound. However, it should be noted that fuzzy inference systems of the class being

designed here are universal approximators [54,23]. Thus, for a sufficiently large number of MFs and rules, any input-output mapping should be approximated with an arbitrary accuracy after the identification and optimization stage, i.e., the training error should be as small as required. In practice, it will be shown that the iterative identification and tuning process proposed here converges fast and the number of MFs required per input is in most cases below 5, with very few exceptions.

## 4 Case Study and Validation: ESTSP 2007 Competition Dataset

For the purposes of validating and illustrating the proposed methodology framework and concrete algorithms and criteria used in this paper, we analyze the data set from the competition of the first European Symposium on Time Series Prediction (ESTSP 2007) [14]. This data set, see figure 2, consists of 875 samples of weekly temperatures of the El Niño-Southern Oscillation phenomenon.

In this section we analyze the original ESTSP 2007 series split into two subsets: a training set (first 475 samples) and a second set (last 400 samples) that will be used for validation. We will call this series ENSO. Though one of the major goals of the proposed methodology is to avoid the requirement of validation and test series, we define two subsets in order to validate the methodology with the residual noise estimator and algorithms being used. In this case study, it will be shown that the delta test as well as the fuzzy systems identification and optimization methods used are appropriate for implementing the proposed methodology framework.

For this case study, a maximum regressor size of 10 and a prediction horizon of 50 are considered, i.e., the last 10 known values will be used for predicting the next 50 (unknown) values. To this end, 50 different fuzzy inference systems have to be built in order to model the dynamics of the system for prediction horizons 1 through 50.

### *4.1 Variable Selection*

Following the flow depicted in figure 1, in the first stage of our methodology, DT is performed on the training set for all the possible variable selections ($2^{10} - 1$) and the one that leads to the lowest DT estimate is chosen. This process is performed independently for each prediction horizon. The number of selected variables is shown in figure 3. Between 3 and 5 variables are selected out of a maximum of 10. Thus, the employment of DT based variable selection leads to a significant decrease of the complexity of the fuzzy inference systems in terms of number of inputs. This fact, in turn, relieves the curse of dimensionality problem. Thus, for a given maximum regressor size, an initial input selection stage allows for a better fitting of the model.

We should note that a maximum regressor size larger than 10 could be considered. When there are enough data samples, a larger regressor size could be expected to provide accuracy improvements. However, for sizes above 15 or 20 approximately, an exhaustive search becomes too computationally expensive and finally unfeasible with current computational resources. A regressor size of 10 has thus been selected as a twofold heuristic compromise. First, the whole space of possible selections can be explored within a reasonable amount of time (approximately 1 hour for 50 models in a current general purpose computer). Second, after variable selection, the number of inputs is sufficiently small so that the curse of dimensionality problem in nonlinear models does not have a severe impact. Larger regressor sizes, for which the DT estimate is lower, usually lead however to little improvement or even poorer performance of the models. We thus, have selected 10 as an initial regressor size. As we will see later on in this section, in the case of fuzzy inference models this leads to systems with a number of inputs and rules sufficiently small so as to be easy to read by humans. The effect that different maximum regressor sizes can have on model performance will be illustrated

with several cases in section 5.

### *4.2 Identification and Tuning*

As second stage, once input variables have been selected, an iterative identification and tuning process is carried out in three substages, as shown in figure 1. In the first substage, the W&M algorithm is applied to the training set in order to identify fuzzy inference systems. These models are then tuned in the second substage through supervised learning using the Levenberg-Marquardt algorithm over the training set. The process is repeated for increasing numbers of linguistic labels (or MFs) per input, starting from 2. Within this iterative process, in the third substage (complexity selection) the DT estimate is used to check whether the best possible approximation has been achieved, i.e., the right compromise between model complexity and training error has been found.

For the horizon 1 regressor, table 1 shows the number of rules identified for different numbers of linguistic labels per input (between 2 and 15). Training and validation errors are shown as well. The two columns labeled "before tuning" show the errors for the fuzzy systems as identified by means of the W&M algorithm, while the columns labeled "after tuning" show the errors for the systems tuned by means of supervised learning.

After the tuning substage, there is a considerable accuracy improvement. In particular, it can be seen that tuned systems with a low number of rules perform better than untuned systems with a much greater complexity. Thus, the supervised learning substage also contributes to reducing model complexity.

We also note that systems with a low number of linguistic labels per input (particularly between 2 and 5) are only rough approximators before tuning. However, after the tuning substage their accuracy is improved significantly while keeping the same rule base. This

fact suggests that the rule bases correctly reflect the underlying dynamics of the series, though tuning the membership function parameters is no doubt required in order to build accurate models with such a low number of linguistic labels.

Within the methodology proposed here, in this case the identification and tuning stage proceeds as follows. First, in the identification substage a fuzzy inference system is identified using the W&M algorithm with 2 MFs per input. The system consists of 6 rules. After the tuning substage, the system yields a training MSE higher than the DT estimate. Thus, this system is rejected in the complexity selection substage. Then, the process is repeated for a system with 3 MFs per input, for which 15 rules are identified. In this case, the system yields an MSE lower than the DT estimate after being tuned. Thus, it qualifies as a proper model in the complexity selection substage and the identification and tuning stage finishes [2] .

### *4.3 Interpretability Issues and Examples*

The number of MFs or linguistic labels defined for each input has significant consequences on accuracy and interpretability. If it is too low, the accuracy of the system will not suffice. If it is high, the linguistic labels will become too specific, and the number of fuzzy rules identified can be overwhelming, since it can grow up to the product of the number of MFs of every input.

Let us now consider the interpretability issues that arise after the tuning substage. Fuzzy inference systems are inherently comprehensible, specially when the rules are defined by human experts. However, when rules are automatically identified from data and optimization methods are applied, as is the case of the proposed methodology,

[2] In practice, as detailed later on in this section, we use a certain error threshold based on the DT estimate rather than the estimate itself.

interpretability cannot be guaranteed in general [7].

Since the identification substage is implemented using the W&M method, the initial input MFs, of Gaussian type, define a grid uniform partition of the input domain. The output MFs, of singleton type. correspond to output values identified on the training data. The W&M method also guarantees the consistency of the rule base. In a system of this kind, linguistic labels can be easily assigned to each input MF by domain experts, or simply by using the common "LOW", "MEDIUM", etc. labels. The meaning of singleton output values is evident as well. In addition, the rules identified are of the *if-then* type with only conjunction operations in the premise. Thus, these systems can be expected to be easy to read and potentially lead to a physical interpretation.

However, the tuning substage consists in changing the parameters of the input and output MFs with the objective of finding the lowest MSE. After the tuning stage, interpretability could be thus severely compromised. A variety of approaches and methods have been proposed to improve or guarantee interpretability to some extent [7]. Nonetheless, in our application it can be found in practice that the parameter changes do not modify the initial uniform partition of the input space to an extent significant for approximate, linguistic human interpretation. On the one hand, changes in the output MFs should not decrease interpretability in an automated methodology. On the other hand, the changes in the input MFs do not modify the initial partition in a severe manner. The shape and distribution of the tuned MFs of the regressor for horizon 1 are shown in figure 4. In this case, the widths change by 12.9% on average, while the centers of the MFs are shifted on average by 3.1% of the range of the series, with respect to the initial grid uniform partition. More general results for the set of series analyzed in this paper are given in appendix B, table 6.

For the 1 step ahead regressor, considering the notation for discrete time series

introduced in section 3, three input variables are selected to predict $y_{t+1}$: $y_t$, $y_{t-2}$ and $y_{t-7}$. This selection indicates that the next weekly temperature depends only on the values of the last week as well as 2 and 7 weeks before. In addition, the relations between the inputs and the output can be interpreted linguistically. Three linguistic terms are defined for each input, as shown in figure 4, that can be thought as "LOW", "MEDIUM" and "HIGH", represented by Gaussian MFs in the range of observed values, $[18.9, 29.2]$. For instance, a rule that has the highest output temperature as consequent reads as follows:

IF $y_{t-7}$ was $\text{HIGH}_1$ AND $y_{t-2}$ was $\text{HIGH}_2$ AND $y_t$ was $\text{HIGH}_3$ THEN $y_{t+1} \leftarrow$ "29.2°C",

where "29.2°C" is used as linguistic label for a singleton output centered at 29.2.

Of course, this example can be regarded only as a simple particular case. The procedure required to provide a physical interpretation of the models is case dependent to a great extent. In the interpretation process, additional techniques for simplification of fuzzy inference systems can be of considerable help [4]. For instance, if the system is pruned in order to keep only the six best rules with respect to the training set, a system with a test MSE only 6.3% higher than that of the original system is obtained.

Let us now illustrate with an example the process of fuzzification, inference and defuzzification. Consider the application of the previous fuzzy rule to a data point in the test series such that $y_{t+1} = 28.8$ has to be predicted from the following previous values: $y_{t-7} = 28.9$, $y_{t-2} = 29.2$, $y_t = 29.1$. Given these input values, the membership degrees for the fuzzy sets involved in the rule premise are $\mu_{HIGH_1}(28.9) = 0.925$, $\mu_{HIGH_2}(29.2) = 0.990$, $\mu_{HIGH_3}(29.1) = 0.997$, and thus the firing degree of the rule is $0.925$, resulting from applying the minimum conjunction operator on the membership degrees of the premise of the rule. This is the firing degree of the rule, and weights the contribution of the rule consequent to the final output of the

system. In this case, it is the rule that activates the most, since the three input values clearly fall within the "HIGH" region of the input space. In order to perform an inference, the 15 rules in the rule base are aggregated using the fuzzy mean defuzzification method, i.e., the singleton conclusions of the rules are averaged using their respective firing degrees as weight over the sum of the 15 firing degrees, according to equation 1, yielding $\hat{y}_{t+1} = 28.728$ as prediction of the inference model for this particular data sample of the ENSO series.

The significant contribution of the application of an input selection stage based on an effective, nonlinear method can be clearly seen in this case. If we use the DT estimate to select the complexity of the system but no input selection is performed, and thus the inference system has 10 inputs, 3 MFs per input and 98 rules are identified, while the test error is 26% higher and the computational cost increases by more than an order of magnitude. If instead only the three last known values are considered as inputs, 8 MFs per input and 65 rules are identified, while the test error is 31% higher. In the latter case, the use of techniques for finding a better embedding delay [25] and selecting the three inputs accordingly do not provide significant improvements.

### *4.4 Model Selection*

Let us now consider the use of the DT estimate for selecting the proper complexity of the tuned fuzzy systems. Rather than using the DT estimate itself, a tolerance band above it is considered. This band is defined by a threshold (DT based threshold, $\text{DTBT}_h$) which increases with increasing horizons $h$ according to equation 2, where $\text{DT}_h$ is the DT estimate for horizon $h$.

$$\text{DTBT}_h = (1 + min(0.90, 0.15 \cdot h)) \cdot \text{DT}_h \tag{2}$$

For each horizon $h$, the simplest system that satisfies $MSE_h \leq DTBT_h$, where $MSE_h$ is the training mean square error, is selected as the best autoregressor. This threshold has been defined on the basis of trial and error as a soft limit that favors simplicity to the detriment of accuracy. However, it was found to be robust for all the series analyzed. The definition is based on the following empirical observations:

- A tolerance of approximately 15% over the DT estimate for horizon 1 is appropriate.
- The best results can be achieved with tolerances increasing with the prediction horizon (particularly for the first 10 predictions approximately).
- A tolerance between 80%-100% over the DT estimates provides good results for long-term prediction.

We note though that the impact of the threshold is not determining for accuracy (the error increase is of the order of 10-20% at most for any prediction horizon). Similar results can be achieved by selecting a fixed adjustment factor of around 50%-75%. We chose the particular values in equation 2 so as to favor model simplicity to the detriment of accuracy. This is further discussed in section 6.

For the ENSO series, $\text{DTBT}_1 \approx 1.26 \cdot 10^{-3}$ and, as shown in figure 5, the fuzzy system with 3 linguistic labels per input is chosen as the best autoregressor for horizon 1.

### *4.5 Results and Accuracy Comparison*

Considering now the performance of our methodology for short- and long-term prediction, figure 6 shows the normalized DT NNE estimates (NDT-NNE) for prediction horizons up to 50 as well as the training and validation errors of the fuzzy autoregressors built.

We note that besides the limitations of the fuzzy modeling techniques being employed,

an additional source of error has been introduced in the proposed methodology: the DT based selection of complexity does not guarantee optimal selection under all conditions. Although the fuzzy regressor for horizon 1 prediction that is chosen is the one with the lowest validation error, this is not the case for all horizons. In general, the deviation from the optimal selection depends on the time series being modeled and the prediction horizon.

Let us now compare the validation errors of the systems actually selected against the lowest validation errors that could have been achieved for any complexity. This way we can know the order of magnitude of the error due to the imperfection of the DT based complexity selection. Figure 7 compares the NDT estimate (a robust estimation of the lowest training error that can be achieved without overfitting), the validation errors of the fuzzy autoregressors selected according to the DT estimate, and the lowest possible validation errors for any number of linguistic labels.

Figure 8 shows the predictions for the first 50 values after the training set together with a fragment of the actual time series.

Finally, we compare the accuracy of fuzzy models against LS-SVM models with the same autoregressor size and input selection. Other models are compared in appendix A. LS-SVMs were built for the same training subset selecting Radial Basis Function (RBF) kernels, grid search as optimization routine and cross-validation as cost function, see [48] for a detailed specification of these and other options. Concrete implementation details will be given in section 6. Figure 9 shows the training and generalization errors for LS-SVM and fuzzy models. Averages errors are listed in table 2. Two main conclusions can be drawn from the comparison:

- As for generalization capability, the performance of fuzzy autoregressors is clearly better than that of LS-SVM models. There are 4 exceptions: test errors of fuzzy

autoregressors are slightly higher (less than 5%) for horizons 14, 18 and 19. However, the overall superiority of fuzzy regressors is clear and specially evident for long-term prediction (beyond horizon 25).

- Training and generalization errors are much closer for fuzzy models than for LS-SVM models. For long-term prediction, generalization errors may be even lower than training errors. Also, generalization errors are within approximately 160% of training errors for the worst cases. Thus, training errors of fuzzy models can be trusted as more realistic estimations of the order of the out-of-sample prediction errors.

## 5 Experimental Results

In this section, the proposed concrete implementation of the methodology framework described is applied to a number of varied time series prediction problems from different fields of application, namely the Poland electricity time series prediction benchmark, the monthly averaged sunspot number, the daily averaged aggregated traffic in the Internet2 backbone network, the laser generated data set of the Santa Fe time series competition, and the Mackey-Glass series. In order to ensure reproducibility, the relevant data sources, methods, software tools and parameters used are specified.

For every series, models are built to predict the next 50 values. Though one of the major goals of the methodology proposed here is to avoid the need for validation and test series, we will split the series into two subsets in order to assess the out-of-sample prediction performance of the methods being used. Following the methodology illustrated in the previous section, we will summarize the results of the input selection stage and the training and test errors. Results will be compared against those of analogous LS-SVM models built using the same input selection scheme, RBF kernels, grid search as optimization routine and cross-validation as cost function. In appendix A

we show a further comparison with other modeling techniques.

### *5.1 Poland Electricity Benchmark*

This time series (PolElec henceforward) represents the normalized average daily electricity demand in Poland in the 1990´s. The benchmark consists of a training set of 1400 samples, shown in figure 10(a), and a test set of 201 samples, shown in figure 10(b). It has been shown that the dynamics of this time series is nearly linear [27]. Besides the yearly periodicity, a clear weekly periodicity can be seen on smaller time scales (see figure 10(b)).

We will show the results obtained for two different maximum regressor sizes: 7 and 14. In both cases, input selection was performed by exhaustive search of the lowest DT estimate. The number of selected variables is shown in figure 11

Training and test errors of a set of fuzzy autoregressors for horizon 1 are shown in figure 12(a) for different numbers of linguistic labels per input, in the case of a maximum regressor size of 7. The regressor with 5 MFs is selected according to the DT based threshold.

Figure 12(b) shows the training and test errors of fuzzy regressors with different numbers of linguistic labels for prediction horizon 7 (also in the case of a maximum regressor size of 7). The system with 2 linguistic labels is selected according to the DT based threshold. However, the system with 3 linguistic labels achieves the lowest test error. This is an illustrative case in which a simpler and less accurate model is selected because of the permissive nature of the DT based threshold. Besides a lower number of linguistic labels, the system with 2 linguistic labels per input has 8 rules, whereas the system with 3 linguistic labels per input has 15 rules.

Figure 13 shows the DT estimates as well as training and test errors for the two regressor sizes considered. The average training and test error of LS-SVM models are shown together with the errors of fuzzy models in table 2. Fuzzy autoregressors achieve a greater approximation accuracy for the test subset. In this case, there are no exceptions for any prediction horizon, and the differences are higher than in the case of the ENSO series. We also note that for this series test errors are bounded within a range of 133% of training errors.

### *5.2 Sunspot Numbers*

The series of sunspot numbers is a periodic measure of the sunspot activity as a function of the number of spots visible on the face of the sun and the number of groups into which they cluster. Values from this series (Sunspots) are subject to uncertainty and noise, particularly during the past centuries. We analyze a series of monthly averaged sunspot numbers covering from January 1749 to December 2007, as provided by the National Geographical Data Center from the US National Oceanic and Atmospheric Administration [3] . The series is split into a set of 1000 values for training and a set of 2908 values for testing. The whole series is shown in figure 14.

Figure 15 shows the number of variables selected for the two maximum regressor sizes considered for the Sunspots series: 9 and 12. Figure 16 shows the DT estimates as well as training and test errors for the two maximum regressor sizes chosen. The average training and test error of LS-SVM models are shown together with the errors of fuzzy models in table 2. For both regressor sizes, fuzzy autoregressors provide more accurate

[3] The series used here can be obtained from http://www.ngdc.noaa.gov/stp/SOLAR/ftpsunspotnumber.html. The International Sunspot Number is produced by the Solar Influence Data Analysis Center (SIDC) at the Royal Observatory of Belgium [49].

out-of-sample predictions with no exception for any of the prediction horizons.

### *5.3 Aggregated Incoming Traffic in the Internet2 Backbone Network*

This series, Internet2 henceforward, represents the total amount of aggregated incoming traffic in the routers of the Abilene network, the Internet2 backbone, during several years. The Internet2 series consists of 1458 daily averages (in bps), shown in figure 17 covering from the 4th of January of 2003 to the 31st of December of 2006. The data are available from the Abilene Observatory [21]. The daily averages for years 2003 and 2004 (the first 728 values) were selected as training set, whereas the daily averages for years 2005 and 2006 (the last 730 values) were selected as test set.

Figure 18 shows the number of variables selected for the two maximum regressor sizes considered for the Internet2 series: 7 and 12. For these two cases, figure 19 shows the DT estimates as well as training and test errors. The average training and test error of LS-SVM models are shown together with the errors of fuzzy models in table 2. Again, for both regressor sizes, fuzzy autoregressors are more accurate with no exception for any of the prediction horizons.

### *5.4 Santa Fe Time Series Competition: Laser Dataset*

The laser data set of the Santa Fe Laser time series competition [45] (SFL) consists of 1000 training samples and 9000 test samples, as shown in figure 20. The series represents the intensity of a far-infrared-laser in a chaotic state, measured in a physics laboratory experiment. This time series is a cross-cut through periodic to chaotic pulsations of the laser. In this case, as opposed to the previous series, the underlying system can be described as a low dimensional deterministic system using three coupled

ordinary differential equations. Indeed, chaotic pulsations can be closely modeled using the theoretical Lorenz model of a two level system [52,25], since the experiment was designed to fulfill the condition of being describable by this model as closely as possible.

This series is a remarkable example of noise-free complicated behavior in a clean, stationary, low-dimensional physical system for which the underlying dynamics is well understood. The data set is very predictable on short time scales because of the relatively simple oscillations. However, the rapid decay of the oscillations are events harder to predict.

In this case, two maximum sizes are considered: 10 (for which exhaustive search of DT estimates is applied) and 16 (for which the exhaustive search is extended with a forward-backward search up to size 16). The number of variables selected for both cases is shown in figure 21.

Figure 22 shows the DT estimates as well as training and test errors for the two regressor sizes considered. As shown in table 2, for this series LS-SVM based autoregressors clearly outperform their fuzzy counterpart in terms of accuracy.

### *5.5 Mackey-Glass Series*

The Mackey-Glass time series [31] (MG henceforth) is another case of fully deterministic dynamics. However, this series is generated numerically. It is often used in the literature for evaluating nonlinear methods and fuzzy systems identification and prediction methods in particular [51,26,44,22,12]. The MG series is defined by the following differential equation:

$$\frac{dy(t)}{dt} = \frac{0.2y(t-\tau)}{1+y^{10}(t-\tau)} - 0.1y(t).$$

When $\tau > 17$, the series exhibits chaotic behavior. Higher values of $\tau$ yield higher dimensional chaos. In this section, a discrete time series is generated using the 4th order Runge-Kutta numerical integration method with $\tau = 30$.

A series of 1500 values (see figure 23) was generated and splitted into a set of 500 samples for training and a set comprising the remaining 1000 samples for test. As in [51], we use a maximum regressor size of 9.

Figure 24 shows the number of selected variables for horizons up to 50. Figure 25 shows the training and test errors together with the DT estimates. From table 2, it is evident that LS-SVM models achieve a greater accuracy averaged for horizons 1 through 50.

For comparison purposes with the literature about fuzzy modeling of the Mackey-Glass series, we consider the 1 step ahead autoregressor for the MG series. For a regressor size of 9, the inference system has only two inputs, both with 5 linguistic labels, and 13 rules. In spite of the simplicity of this system, its test error is approximately 9% lower than the DT estimate.

## 6 Discussion

In this study, no preprocessing stage has been performed on the datasets. Preprocessing techniques, such as detrending, rescaling, seasonal adjustment, noise reduction, and wavelet decomposition, among many others [9,25], can be useful depending on the dataset characteristics and particular field of application. The extent to which predictions can be improved with preprocessing techniques is however difficult to quantify in general, specially when nonlinear models are used. In particular, it is worth to mention that the presence of outliers can have a significant impact on the results [10,9]. First, the modeling techniques used here rely on the MSE as an error measure and are thus

inherently sensitive to outliers. Furthermore, the results from the DT are also sensitive to outliers, both for input selection and residual variance estimation. Therefore, the presence of outliers should not be disregarded in a general context, and outlier detection algorithms [10,9] should be employed in the preprocessing stage in applications where outliers are involved.

The combined use of a nonparametric noise estimation method with fuzzy modeling techniques has been experimentally shown to perform well for long-term time series prediction. The methodology developed does not require a validation stage and thus the whole available data set can be used as training data to build autoregressive models.

The use of an a priori approach for both variable selection and structure selection drastically reduces the computational cost. Furthermore, the use of DT estimates in a first input selection stage as well as in the identification and tuning stage has been shown to be advantageous in two main aspects:

- The use of DT for input selection improves the interpretability of the fuzzy models built since only the relevant variables are inputs to the inference systems. This fact, in turn, greatly simplifies the whole structure of the inference system and alleviates the curse of dimensionality problem. Input selection allows for a drastic reduction of the number of inputs. For instance, this is specially clear for the MG series, for which only 2 inputs out of 9 are selected for short-term prediction (horizons 1 through 4).
- It has been shown to be a robust solution to the problem of selecting the proper system complexity, providing satisfactory performance for heterogeneous experimental data.

In general, the optimal $\text{DTBT}_h$ threshold in terms of accuracy is dependent on the data set, the nonlinear approximation technique as well as the particular parameters employed, i.e., fuzzy operators, MFs, inference model, and the identification and tuning methods. The definition of a particular threshold can be thus understood as a hint on what

degree of accuracy is expected when a particular fuzzy modeling technique is applied.

In this paper, a tolerant DT based threshold has been defined. With a more strict threshold, more accurate models could be built. However, by using a tolerant DT based threshold, we have favored simplicity to the detriment of accuracy. This way, linguistic interpretation of the models is easier and we can thus exploit in practice a fundamental advantage of fuzzy inference models.

All these factors contribute to a methodology for building fuzzy inference models that are both accurate and interpretable for both short-term and long-term prediction. In addition, fuzzy models have been shown to clearly outperform LS-SVM models in terms of prediction accuracy in the case of noisy time series for which there are no satisfactory deterministic models available. For the series shown in table 2 and excluding the SFL and MG series, the average test error of LS-SVM models is 52% higher than that of fuzzy models. For further comparison with other, less accurate modeling techniques, results for OP-ELM and $k$-NN models are given in appendix A.

A remarkable property of the fuzzy regressors developed with our methodology is their generalization capability. Test errors have been found to be very close to training errors. The difference between them is typically no more than 20-30% except in the case of the Sunspots series, where test errors are approximately 60% higher than training errors. While LS-SVM are usually praised for their good generalization performance, fuzzy autoregressors exhibit a much lower degree of overfitting.

On the other hand, It has been shown that LS-SVM models achieve a greater accuracy than fuzzy models for a specific type of series represented by the SFL and MG series. Both are noise-free, stationary, low-dimensionally chaotic, can be predicted with relatively simple analytical models and can be approximated with a very high accuracy. Similar results can be obtained for a wide range of series of the same class. This fact

leads us to conclude that in the absence of noise and perturbations, fuzzy inference based autoregression may not be a proper technique if the main objective is approximation accuracy and interpretability is a secondary objective. This type of series is however not common in many real world applications. In addition, the higher accuracy of LS-SVM does not come at no cost. For the MG (9), SFL (10), and SFL (16) series, the construction and optimization of the LS-SVM model requires approximately 37, 35 and 103 times more run time respectively, as shown in table 3. Thus, the methodology proposed here, while clearly less accurate for this kind of series, is still significantly faster and exhibits less overfitting.

As far as computational requirements is concerned, the proposed methodology has a very low cost compared against the LS-SVM method. This factor has important practical implications that are often neglected or only partially addressed in the literature. In fact, the high and often unaffordable cost of the LS-SVM and other accurate modeling techniques has recently motivated the development of faster nonlinear learning machines for time series applications [47,20].

A tool, xftsp [38], has been developed that implements the methodology proposed in this paper and provides support for the identification and tuning algorithms included in the Xfuzzy development environment for fuzzy systems [53]. The design of the xftsp tool allows for the use of the wide set of tools available in the Xfuzzy environment for complementary tasks such as visualization, simplification and code generation. This Java based implementation of the methodology presented here is consistently between 1 and 2 orders of magnitude faster than the implementation of LS-SVM used for this study: the optimized C version of the LS-SVMlab1.5 Matlab/C toolbox [30]. Table 3 shows the time required to build models with both methods for a subset of the time series considered in this paper. Memory consumption is also much lower for the fuzzy methodology, which enables it to be applied to large training series beyond the few

thousand samples current practical limitation of LS-SVM models.

The fact that the test results are improved when a DT based threshold higher than the DT estimate itself is introduced, leads us to two remarks on the performance of the identification and tuning stage:

- There is likely room for improvement of the identification and tuning procedures.
- The DT based threshold can be seen as an aggressiveness index. 1 would be the most aggressive option, most often leading to overfitting and high complexity. Values in the range $[1.2, 2]\text{DT}_h$ are reasonable for the identification and learning techniques employed, most often leading to both low complexity and overfitting.

The fact that the impact of the DT based threshold is very similar for all the series analyzed leads us to conclude that it is a factor eminently dependent on the identification and learning procedure and its inner limitations. Other methods for fuzzy inference systems identification, tuning and simplification exist, and the ones used in this paper could be improved. This is an area of future research.

For complex and noisy time series, it is common that the most simple fuzzy system that can be built (the one with 2 linguistic labels per input) is comparable in accuracy to the LS-SVM model. For example, the fuzzy system with 2 linguistic labels per input for horizon 1 prediction of the PolElec series outperforms LS-SVM with the same input selection. In this case, the test error of the fuzzy regressor is approximately 35% lower.

In general, it can be concluded that fuzzy systems with the minimum number of linguistic terms, though not optimal in terms of accuracy, provide a reasonable approximation to the best system that can be built. Thus, it is easy to obtain very simple approximate models that ease the understanding of the time series dynamics.

## 7 Conclusion

We have proposed an automatic methodology framework for long-term time series prediction by means of fuzzy inference systems. The use of a nonlinear input selection method yields improved accuracy and interpretability. Experimental results for a concrete implementation of the methodology confirm the satisfactory approximation accuracy and generalization capability of fuzzy regressors. Linguistic interpretability and significantly lower computational requirements are two remarkable advantages over common time series prediction methods.

A fundamental advantage of autoregressive time series prediction with fuzzy inference systems is the fact that the models constructed consist of linguistic rules that can be interpreted by humans. For some time series, the most accurate rule bases have a low number of rules (below 10-15 rules), making it easy to draw a linguistic explanation of the system dynamics.

Several procedures have been shown to play a key role in achieving good approximation accuracy and low overfitting while keeping the complexity low: variable selection, application of a supervised learning method for tuning after identification, and using DT-NNE for selecting the proper number of linguistic labels per input. Also, when systems have a high number of rules and are thus not interpretable by humans in practice, there is still the possibility to build simpler, approximate models with a degree of accuracy of the same order.

The proposed methodology has been shown to clearly outperform LS-SVM based predictions in terms of approximation accuracy except in the particular case of noise-free, stationary and deterministic time series, where fuzzy autoregressors are still significantly faster to build and exhibit less overfitting.

## A Accuracy Comparison of Different Methods

For further comparison, the modeling and prediction accuracy of different modeling techniques for the series analyzed in this paper are shown in table 4. Two alternatives are considered in addition to LS-SVM: OP-ELM and $k$-NN models. Errors are shown in units relative to that of fuzzy models.

The extreme learning machine (ELM) [20] is a simple yet effective learning algorithm for training single-hidden-layer feed-forward artificial neural networks with random hidden nodes. OP-ELM [47] is a methodology based on the ELM, that has been shown to produce models competitive against well-known, accurate techniques, such as LS-SVM and Multilayer Perceptron, while being significantly faster. OP-ELM models were built using the OP-ELM toolbox [33] with the following configuration options: a combination of linear, Gaussian and sigmoid kernels, a maximum of 100 neurons, and data normalization before modeling. $k$-NN models were generated with 10 as maximum number of neighbors, using the Euclidean distance and 10-fold cross-validation for selecting the best $k$. LS-SVM models were generated as detailed in sections 5 and 6.

Considering the out-of-sample or test error, LS-SVM and OP-ELM are in general more accurate than $k$-NN models, with only a slight exception for the SFL (16) series. LS-SVM are in general more accurate than OP-ELM models, with the clear exceptions of the Sunspots (9) and Sunspots (12) series, where OP-ELM models are the most accurate and also outperform fuzzy models. OP-ELM are also slightly more accurate than LS-SVM for the AbileneI (7) series and slightly less accurate for the AbileneI (14). It should be noted that the Sunspots series is considerably nonstationary and the test set differs from the training set significantly.

## B Comparison of Different Neuro-Fuzzy Methods

In the particular implementation of the proposed methodology used in this paper, the W&M and Levenberg-Marquardt methods have been employed for identification and tuning. This selection has been made on the basis of their satisfactory results in terms of both accuracy and interpretability. Here we compare the performance of these two methods against some other alternatives.

Table 5 compares the test errors obtained with different methods for identification and tuning of fuzzy inference systems. For easier comparison, errors are shown relative to the errors of the inference models built using the W&M and Levenberg-Marquardt methods within the methodology proposed in this paper. For instance, the 50 fuzzy inference models for the ENSO series, when identified using the W&M method and optimized using the Rprop method, have an average test error 3.8% higher than that shown in section 5, table 2 for the W&M and Levenberg-Marquardt methods.

Two identification methods are considered in the table: W&M and the method based on subtractive clustering (SC) proposed by Chiu [11]. The W&M method was found to consistently provide better results than other grid partition based methods, such as the algorithm by Higgins and Goodman [18]. The SC method was found to consistently provide better accuracy than other clustering based identification alternatives using the Gath-Geva [1,15], Gustafson-Kessel [17], hard and fuzzy C-means [13] clustering methods.

A number of supervised learning algorithms were tested for the tuning substage. For this study, we used the implementations in the Xfuzzy environment, see [39] for a more detailed description of the wide range of methods supported. Among them, we distinguish four classes of methods: gradient descent [32], conjugate gradient, second

order or quasi-Newton [3], and algorithms with no derivatives. Table 5 shows the test errors for the best option from each of the first three classes of algorithms: Resilient Propagation (Rprop) [42,32], from the gradient descent class, Scaled Conjugated Gradient (SCG) [35], from the conjugate gradient class, and Levenberg-Marquardt (L-M) [3], from the second order class of methods. The following parameters were used for the L-M method: initial Hessian addition 0.1, increase factor 10.0 and decrease factor 0.2. It is worth to mention that none of the statistical or probabilistic methods was found to be competitive in terms of performance, being unable to achieve training errors below the DT based threshold in most cases, within reasonable time bounds. These include the Simulated Annealing method with different cooling schemes, Downhill Simplex and Powell´s methods [39]. We note however that these methods are highly dependent on the values of several parameters that could be explored only partially.

From table 5, it is clear that the most accurate models can be obtained with the W&M identification method. Regarding the tuning method, the SCG method is only slightly less accurate than the L-M method, while the Rprop method is slightly more accurate than L-M. The three options have been selected as the most accurate among the set of methods tested, achieving in practice very similar results in terms of accuracy.

However, one of the main objectives of this study is to build models that are accurate, yet as simple as possible. If we look at the number of MFs required to achieve such degrees of accuracy, the L-M method is more efficient, as can be seen in table 6. The table shows three measures of complexity of the fuzzy systems generated with each method: the number of MFs, as well as the percent shift of the centers and the percent change of widths of the MFs after tuning, with respect to the initial uniform grid partition. For a lower number of input MFs, higher changes in the shapes of the MFs can be expected.

As a conclusion, we used the L-M method in this work because it produces systems with

the lowest number of MFs, while being almost as accurate as possible with the methods tested. We note however that the Rprop method is almost equivalent in terms of the number of MFs required. Rprop would be thus a good alternative, yielding models that are slightly more complex and accurate.

**Acknowledgements**

The first author is supported by a Marie Curie Intra-European Fellowship for Career Development (grant agreement PIEF-GA-2009-237450) within the European Community´s Seventh Framework Programme (FP7/20072013). Most of this work was done while the first author was with the Microelectronics Institute of Seville, IMSE-CNM, CSIC – Scientific Research Council. C. Americo Vespucio s/n. Parque Tecnologico Cartuja. E-41092 Seville, Spain.

This work has been supported in part by project TEC2008-04920/MICINN from the Spanish Ministry of Science and Innovation, as well as project P08-TIC-03674 and grants IAC07-I-0205:33080 and IAC08-II-3347:56263 from the Andalusian regional Government.

## References


[1] J. Abonyi, R. Babuska, F. Szeifert, Modified Gath-Geva Fuzzy Clustering for Identification of Takagi-Sugeno Fuzzy Models, IEEE Transactions on Systems, Man and Cybernetics, Part B 32 (5) (2002) 612–621.

[2] P. P. Angelov, D. P. Filev, An approach to online identification of takagi-sugeno fuzzy models, IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics 34 (1) (2004) 484–498.

[3] R. Battiti, First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method, Neural Computation 4 (2) (1992) 141–166.

[4] I. Baturone, F. J. Moreno-Velo, A. Gersnoviez, A CAD Approach to Simplify Fuzzy System Descriptions, in: 15th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'06), 2006.

[5] R. E. Bellman, Dynamic Programming, 1957th ed., Republished, Dover Publications Inc., 2003, ISBN: 0486428095.

[6] G. Box, G. M. Jenkins, G. Reinsel, Time Series Analysis: Forecasting & Control, Prentice Hall; 3rd edition, 1994, ISBN: 0130607746.

[7] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (eds.), Interpretability Issues in Fuzzy Modeling, Studies in Fuzziness and Soft Computing, Springer Verlag, Berlin, Germany, 2003, ISBN: 978-3-540-02932-8.

[8] Y.-H. O. Chang, B. M. Ayyub, Fuzzy regression methods - a comparative assessment, Fuzzy Sets and Systems 119 (2) (2001) 187–203.

[9] C. Chatfield, The Analysis of Time Series. An Introduction, CRC Press, 2003, Sixth edition, ISBN: 1-58488-317-0.

[10] C. Chen, L.-M. Liu, Joint Estimation of Model Parameters and Outlier Effects in Time Series, Journal of the American Statistical Association 88 (421) (1993) 284–297.

[11] S. L. Chiu, A Cluster Estimation Method with Extension to Fuzzy Model Identification, in: IEEE Conference on Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence, Orlando, FL, USA, 1994.

[12] S. L. Chiu, Fuzzy Model Identification Based on Cluster Estimation, Journal of Intelligent & Fuzzy Systems 2 (3) (1994) 267–278.

[13] J. V. de Oliveira, W. Pedrycz (eds.), Advances in Fuzzy Clustering and its Applications, John Wiley & Sons, Ltd., West Sussex, England, 2007, ISBN: 978-0-470-02760-8.

[14] ESTSP´07 European Symposium on Time Series Prediction: Prediction Competition (Mar. 2008).
URL http://www.estsp.org

[15] I. Gath, A. B. Geva, Unsupervised Optimal Fuzzy Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (7) (1989) 773–780.

[16] S. Guillaume, Designing fuzzy inference systems from data: An interpretability-oriented review, IEEE Transactions on Fuzzy Systems 9 (3) (2001) 426–443.

[17] E. E. Gustafson, W. C. Kessel, Fuzzy Clustering with a Fuzzy Covariance Matrix, in: 17th Symposium on Adaptive Processes, 1978 IEEE Conference on Decision and Control , San Diego, CA, 1978.

[18] C. M. Higgins, R. M. Goodman, Fuzzy Rule-Based Networks for Control, IEEE Transactions on Fuzzy Systems 2 (1) (1994) 82–88.

[19] T.-P. Hong, C.-Y. Lee, Induction of fuzzy rules and membership functions from training examples, Fuzzy Sets and Systems 84 (1) (1996) 33–47.

[20] G.-B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.

[21] The Internet2 Observatory (Jul. 2008).
URL http://www.internet2.edu/observatory/

[22] J.-S. R. Jang, ANFIS: Adaptive-Network-Based Fuzzy Inference System, IEEE Transactions on Systems, Man and Cybernetics 23 (3) (1993) 665–685.

[23] J.-S. R. Jang, C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing A Computational Approach to Learning and Machine Intelligence, Prentice Hall, Upper Saddle River, New Jersey, 1997, ISBN 0-13-261066-3.

[24] A. J. Jones, New Tools in Non-linear Modelling and Prediction, Computational Management Science 2 (1) (2004) 109–149.

[25] H. Kantz, T. Schreiber, Nonlinear Time Series Analysis, 2nd ed., Cambridge University Press, Cambridge, UK, 2004.

[26] N. K. Kasabov, Q. Song, DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 144–154.

[27] A. Lendasse, J. Lee, V. Wertz, M. Verleyssen, Forecasting Electricity Consumption using Nonlinear Projection and Self-Organizing Maps, Neurocomputing 48 (1) (2002) 299–311.

[28] E. Liitiäinen, A. Lendasse, F. Corona, Non-parametric Residual Variance Estimation in Supervised Learning, in: IWANN 2007, International Work-Conference on Artificial Neural Networks, San Sebastián, Spain, 2007.

[29] E. Liitiäinen, A. Lendasse, F. Corona, Bounds on the mean power-weighted nearest neighbour distance, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 464 (2097) (2008) 2293–2301.

[30] Least Squares - Support Vector Machines Matlab/C Toolbox (Apr. 2008).
URL http://www.esat.kuleuven.ac.be/sista/lssvmlab

[31] M. C. Mackey, L. Glass, Oscillations and Chaos in Physiological Control Systems, Science 197 (4300) (1977) 287–289.

[32] G. D. Magoulas, M. N. Vrahatis, G. S. Androulakis, Improving the Convergence of the Backpropagation Algorithm Using Learning Rate Adaptation Methods, Neural Computation (1999) 1769–1796.

[33] Y. Miche, A. Sorjamaa, A. Lendasse, OP-ELM: Theory, Experiments and a Toolbox, in: 18th International Conference on Artificial Neural Networks (ICANN), vol. 5163 of Lecture Notes in Computer Science, Prague, Czech Republic, 2008.

[34] S. Mitra, Y. Hayashi, Neuro-fuzzy rule generation: survey in soft computing framework, IEEE Transactions on Neural Networks 11 (3) (2000) 748–768.

[35] M. F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (4) (1993) 525–533.

[36] F. Montesino-Pouzols, A. Barriga, Regressive fuzzy inference models with clustering identification: Application to the ESTSP08 competition, in: 2nd European Symposium on Time Series Prediction, Porvoo, Finland, 2008.

[37] F. Montesino-Pouzols, A. Lendasse, A. Barriga, Fuzzy Inference Based Autoregressors for Time Series Prediction Using Nonparametric Residual Variance Estimation, in: 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'08), IEEE World Congress on Computational Intelligence, Hong Kong, China, 2008.

[38] F. Montesino-Pouzols, A. Lendasse, A. Barriga, xftsp: a Tool for Time Series Prediction by Means of Fuzzy Inference Systems, in: 4th IEEE International Conference on Intelligent Systems (IS´08), Varna, Bulgaria, 2008.

[39] F. J. Moreno-Velo, I. Baturone, A. Barriga, S. Sánchez-Solano, Automatic Tuning of Complex Fuzzy Systems with Xfuzzy, Fuzzy Sets and Systems 158 (18) (2007) 2026–2038.

[40] F. J. Moreno-Velo, I. Baturone, S. Sánchez-Solano, A. Barriga, Rapid Design of Fuzzy Systems With Xfuzzy, in: 12th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'03), St. Louis, MO, USA, 2003.

[41] H. Pi, C. Peterson, Finding the embedding dimension and variable dependencies in time series, Neural Computation 6 (3) (1994) 509–520.

[42] M. Riedmiller, Advanced supervised learning in multi-layer perceptrons - from backpropagation to adaptive learning algorithms, Computer Standards and Interfaces 16 (3) (1994) 265–278.

[43] I. Rojas, H. Pomares, J. Ortega, A. Prieto, Self-Organized Fuzzy System Generation from Training Examples, IEEE Transactions on Fuzzy Systems 8 (1) (2000) 23–36.

[44] H.-J. Rong, N. Sundararajan, G.-B. Huang, P. Saratchandran, Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction, Fuzzy Sets and Systems 157 (9) (2006) 1260–1275.

[45] The Santa Fe Time Series Competition Data. Data Set A: Laser generated data (Apr. 2008). URL
http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html

[46] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, A. Lendasse, Methodology for Long-Term Prediction of Time Series, Neurocomputing 70 (16–18) (2007) 2861–2869.

[47] A. Sorjamaa, Y. Miche, R. Weiss, A. Lendasse, Long-Term Prediction of Time Series using NNE-based Projection and OP-ELM, in: 2008 International Joint Conference on Neural Networks (IJCNN 2008), IEEE World Congress on Computational Intelligence, Hong Kong, China, 2008.

[48] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, Singapore, 2002, ISBN: 981-238-151-1.

[49] R. A. M. Van der Linden, the SIDC Team, Online Catalogue of the Sunspot Index, RWC Belgium, World Data Center for the Sunspot Index, Royal Observatory of Belgium, years 1748-2007, http://sidc.oma.be/html/sunspot.html (Jan. 2008).

[50] L. X. Wang, The WM Method Completed: A Flexible System Approach to Data Mining, IEEE Transactions on Fuzzy Systems 11 (6) (2003) 768–782.

[51] L. X. Wang, J. M. Mendel, Generating Fuzzy Rules by Learning from Examples, IEEE Transactions on Systems, Man, and Cybernetics 22 (4) (1992) 1414–1427.

[52] A. Weigend, N. Gershenfeld, Times Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley Publishing Company, 1994, ISBN: 0201626020.

[53] Xfuzzy: Fuzzy Logic Design Tools (Aug. 2008).
URL https://forja.rediris.es/projects/xfuzzy

[54] H. Ying, Sufficient Conditions on Uniform Approximation of Multivariate Functions by General TakagiSugeno Fuzzy Systems with Linear Rule Consequent, IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans 28 (4) (1998) 515–520.

## List of Tables

Table 1
ENSO series: number of membership functions and rules as well as errors for prediction horizon 1. Exhaustive DT based selection of inputs. All errors are given as NMSE.

| | | Before tuning | | After tuning | |
|---|---|---|---|---|---|
| #MF | #Rules | Training | Validation | Training | Validation |
| 2 | 6 | $2.833 \cdot 10^{-2}$ | $2.899 \cdot 10^{-2}$ | $1.479 \cdot 10^{-3}$ | $1.705 \cdot 10^{-3}$ |
| 3 | 15 | $8.813 \cdot 10^{-3}$ | $1.016 \cdot 10^{-2}$ | $1.250 \cdot 10^{-3}$ | $1.558 \cdot 10^{-3}$ |
| 4 | 20 | $4.190 \cdot 10^{-3}$ | $4.884 \cdot 10^{-3}$ | $1.189 \cdot 10^{-3}$ | $1.580 \cdot 10^{-3}$ |
| 5 | 31 | $2.709 \cdot 10^{-3}$ | $3.113 \cdot 10^{-3}$ | $1.082 \cdot 10^{-3}$ | $1.616 \cdot 10^{-3}$ |
| 6 | 44 | $1.986 \cdot 10^{-3}$ | $2.466 \cdot 10^{-3}$ | $1.009 \cdot 10^{-3}$ | $1.738 \cdot 10^{-3}$ |
| 7 | 56 | $1.868 \cdot 10^{-3}$ | $2.617 \cdot 10^{-3}$ | $9.228 \cdot 10^{-4}$ | $1.794 \cdot 10^{-3}$ |
| 8 | 66 | $1.453 \cdot 10^{-3}$ | $1.978 \cdot 10^{-3}$ | $9.509 \cdot 10^{-4}$ | $1.869 \cdot 10^{-3}$ |
| 9 | 85 | $1.289 \cdot 10^{-3}$ | $1.915 \cdot 10^{-3}$ | $8.676 \cdot 10^{-4}$ | $1.979 \cdot 10^{-3}$ |
| 10 | 101 | $1.229 \cdot 10^{-3}$ | $1.920 \cdot 10^{-3}$ | $7.509 \cdot 10^{-4}$ | $2.153 \cdot 10^{-3}$ |
| 11 | 128 | $1.130 \cdot 10^{-3}$ | $2.043 \cdot 10^{-3}$ | $6.104 \cdot 10^{-4}$ | $2.602 \cdot 10^{-3}$ |
| 12 | 132 | $1.114 \cdot 10^{-3}$ | $2.113 \cdot 10^{-3}$ | $5.848 \cdot 10^{-4}$ | $2.491 \cdot 10^{-3}$ |
| 13 | 175 | $1.121 \cdot 10^{-3}$ | $2.139 \cdot 10^{-3}$ | $4.902 \cdot 10^{-4}$ | $2.816 \cdot 10^{-3}$ |
| 14 | 178 | $1.006 \cdot 10^{-3}$ | $2.194 \cdot 10^{-3}$ | $4.426 \cdot 10^{-4}$ | $3.455 \cdot 10^{-3}$ |
| 15 | 191 | $9.713 \cdot 10^{-4}$ | $2.126 \cdot 10^{-3}$ | $4.793 \cdot 10^{-4}$ | $2.865 \cdot 10^{-3}$ |

Table 2
Training and test errors of LS-SVM and fuzzy models averaged for prediction horizons 1 through 50. All errors are given as NMSE. Maximum regressor size specified between parenthesis.

| | LS-SVM | | Fuzzy inference | |
|---|---|---|---|---|
| Series | Training | Test | Training | Test |
| ENSO (10) | $8.055 \cdot 10^{-3}$ | $3.192 \cdot 10^{-2}$ | $1.943 \cdot 10^{-2}$ | $2.043 \cdot 10^{-2}$ |
| PolElec (7) | $1.158 \cdot 10^{-2}$ | $3.566 \cdot 10^{-2}$ | $1.696 \cdot 10^{-2}$ | $1.779 \cdot 10^{-2}$ |
| PolElec (14) | $1.037 \cdot 10^{-2}$ | $3.241 \cdot 10^{-2}$ | $1.582 \cdot 10^{-2}$ | $1.816 \cdot 10^{-2}$ |
| Sunspots (9) | $1.338 \cdot 10^{-2}$ | $3.284 \cdot 10^{-2}$ | $1.691 \cdot 10^{-2}$ | $2.623 \cdot 10^{-2}$ |
| Sunspots (12) | $9.637 \cdot 10^{-3}$ | $3.024 \cdot 10^{-2}$ | $1.590 \cdot 10^{-2}$ | $2.546 \cdot 10^{-2}$ |
| AbileneI (7) | $8.587 \cdot 10^{-3}$ | $2.476 \cdot 10^{-2}$ | $1.448 \cdot 10^{-2}$ | $1.732 \cdot 10^{-2}$ |
| AbileneI (12) | $6.771 \cdot 10^{-3}$ | $2.153 \cdot 10^{-2}$ | $1.228 \cdot 10^{-2}$ | $1.506 \cdot 10^{-2}$ |
| SFL (10) | $1.481 \cdot 10^{-3}$ | $6.578 \cdot 10^{-3}$ | $1.020 \cdot 10^{-2}$ | $1.285 \cdot 10^{-2}$ |
| SFL (16) | $5.275 \cdot 10^{-4}$ | $5.290 \cdot 10^{-3}$ | $8.791 \cdot 10^{-3}$ | $1.202 \cdot 10^{-2}$ |
| MG (9) | $7.881 \cdot 10^{-4}$ | $3.658 \cdot 10^{-3}$ | $1.385 \cdot 10^{-2}$ | $1.775 \cdot 10^{-2}$ |

Table 3
Run time (in seconds) required to build models for prediction horizons 1-50. All tests were run on the same system, with no significant competing load. Maximum regressor size specified between parenthesis.

| Series | LS-SVMlab1.5 | Fuzzy inference |
|---|---|---|
| ENSO (10) | $3.45 \cdot 10^5$ | $1.05 \cdot 10^4$ |
| PolElec (7) | $3.04 \cdot 10^5$ | $1.05 \cdot 10^4$ |
| PolElec (14) | $9.91 \cdot 10^5$ | $2.30 \cdot 10^4$ |
| Sunspots (9) | $3.10 \cdot 10^5$ | $1.04 \cdot 10^4$ |
| Sunspots (12) | $2.42 \cdot 10^5$ | $1.22 \cdot 10^4$ |
| AbileneI (7) | $1.40 \cdot 10^5$ | $1.75 \cdot 10^3$ |
| AbileneI (12) | $1.27 \cdot 10^5$ | $4.69 \cdot 10^3$ |
| SFL (10) | $1.28 \cdot 10^6$ | $3.49 \cdot 10^4$ |
| SFL (16) | $1.61 \cdot 10^6$ | $4.55 \cdot 10^4$ |
| MG (9) | $3.64 \cdot 10^5$ | $3.54 \cdot 10^3$ |

Table 4
Accuracy comparison of different nonlinear modeling techniques. Training and test errors, averaged for horizons 1 through 50, of different nonlinear models. Training and test errors are expressed relative to the training and test errors, respectively, of fuzzy models built using the W&M and Levenberg-Marquardt methods within the proposed methodology. Absolute errors for the fuzzy and LS-SVM models were given in section 5, table 2.

| | LS-SVM | | OP-ELM | | $k$-NN | |
|---|---|---|---|---|---|---|
| Series | Training | Test | Training | Test | Training | Test |
| ENSO (10) | 0.41 | 1.56 | 0.76 | 1.91 | 0.18 | 2.30 |
| PolElec (7) | 0.68 | 1.99 | 0.89 | 2.52 | 0.39 | 2.87 |
| PolElec (14) | 0.66 | 1.78 | 0.71 | 2.29 | 0.34 | 2.86 |
| Sunspots (9) | 0.79 | 1.25 | 0.92 | 0.88 | 0.22 | 1.34 |
| Sunspots (12) | 0.61 | 1.19 | 0.90 | 0.93 | 0.13 | 1.66 |
| AbileneI (7) | 0.59 | 1.43 | 0.74 | 1.39 | 0.44 | 2.32 |
| AbileneI (12) | 0.55 | 1.43 | 0.85 | 1.51 | 0.29 | 2.06 |
| SFL (10) | 0.15 | 0.51 | 1.23 | 1.14 | 0.27 | 1.26 |
| SFL (16) | 0.06 | 0.45 | 1.56 | 1.18 | 0.22 | 1.16 |
| MG (9) | 0.06 | 0.21 | 0.25 | 1.52 | 0.19 | 1.63 |

Table 5
Accuracy comparison of different methods for building fuzzy inference models. Test errors, averaged for horizons 1 through 50, are expressed relative to the test errors of the equivalent fuzzy models built using the W&M and Levenberg-Marquardt methods within the proposed methodology. Results are shown for two identification methods: W&M and Subtractive Clustering (SC) based, as well as three tuning methods: Levenberg-Marquardt (L-M), Scaled Conjugated Gradient (SCG), and Resilient Propagation (Rprop).

| | W&M | | | SC | |
|---|---|---|---|---|---|
| | SCG | Rprop | L-M | SCG | Rprop |
| ENSO (10) | 1.003 | 1.038 | 1.120 | 1.065 | 1.249 |
| PolElec (7) | 0.958 | 0.991 | 1.035 | 1.062 | 1.058 |
| PolElec (14) | 0.941 | 0.964 | 1.045 | 1.085 | 1.070 |
| Sunspots (9) | 1.038 | 0.997 | 1.139 | 1.132 | 1.148 |
| Sunspots (12) | 0.996 | 0.953 | 1.068 | 1.100 | 1.061 |
| AbileneI (7) | 1.004 | 0.922 | 2.258 | 2.739 | 1.080 |
| AbileneI (12) | 1.157 | 1.112 | 3.177 | 3.092 | 1.381 |
| SFL (10) | 1.000 | 0.988 | 1.055 | 1.169 | 1.102 |
| SFL (16) | 0.995 | 0.982 | 1.089 | 1.183 | 1.215 |
| MG (9) | 1.051 | 1.011 | 1.082 | 1.184 | 1.123 |
| Average | 1.014 | 0.997 | 1.407 | 1.481 | 1.153 |

Table 6
Complexity comparison of different methods for building fuzzy inference models. Three supervised learning methods are compared: Levenberg-Marquardt (L-M), Scaled Conjugated Gradient (SCG), and Resilient Propagation (Rprop). Three measures of complexity and interpretability are shown: number of membership functions (#MFs), percent center shift ($\Delta c_{j,k}$), and percent width change ($\Delta \sigma_{j,k}$).

| | L-M | | | SCG | | | Rprop | | |
|---|---|---|---|---|---|---|---|---|---|
| | #MF | $\Delta c_{j,k}$ | $\Delta \sigma_{j,k}$ | #MF | $\Delta c_{j,k}$ | $\Delta \sigma_{j,k}$ | #MF | $\Delta c_{j,k}$ | $\Delta \sigma_{j,k}$ |
| ENSO (10) | 3.84 | 8.53 | 21.7 | 4.26 | 6.35 | 21.2 | 4.18 | 7.40 | 18.0 |
| PolElec (7) | 3.08 | 12.3 | 13.2 | 3.14 | 6.11 | 8.82 | 3.10 | 11.0 | 10.3 |
| PolElec (14) | 3.10 | 12.8 | 15.8 | 3.08 | 5.38 | 9.12 | 3.20 | 11.4 | 10.7 |
| Sunspots (9) | 3.02 | 7.40 | 9.72 | 3.02 | 4.66 | 7.14 | 3.00 | 5.89 | 9.72 |
| Sunspots (12) | 3.03 | 9.20 | 12.1 | 3.03 | 5.64 | 8.04 | 3.02 | 6.25 | 10.2 |
| AbileneI (7) | 3.13 | 5.87 | 9.72 | 3.40 | 4.02 | 8.88 | 3.12 | 6.60 | 8.94 |
| AbileneI (12) | 4.60 | 5.13 | 10.9 | 5.26 | 4.99 | 15.4 | 4.44 | 6.73 | 15.2 |
| SFL (10) | 5.03 | 6.03 | 26.5 | 5.94 | 5.26 | 27.8 | 4.97 | 6.81 | 22.7 |
| SFL (16) | 4.77 | 5.88 | 26.1 | 5.36 | 5.31 | 26.4 | 4.71 | 6.57 | 22.4 |
| MG (9) | 3.72 | 12.1 | 17.7 | 4.34 | 7.40 | 16.0 | 4.06 | 10.8 | 16.0 |
| Average | 3.732 | 8.52 | 16.3 | 4.832 | 5.53 | 14.89 | 3.784 | 7.95 | 14.42 |

**List of Figures**

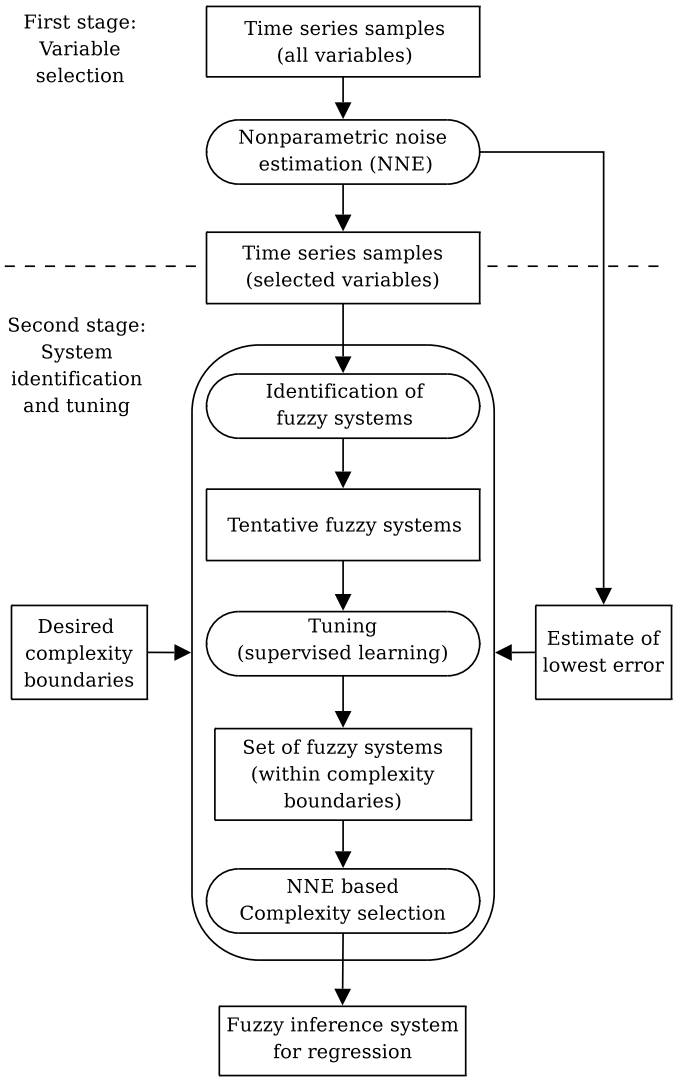



Fig. 1. Methodology Framework for Time Series Prediction.

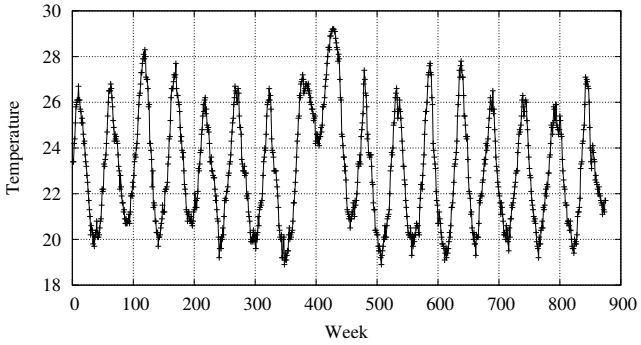



Fig. 2. ESTSP´07 competition data set (ENSO series, 875 samples).

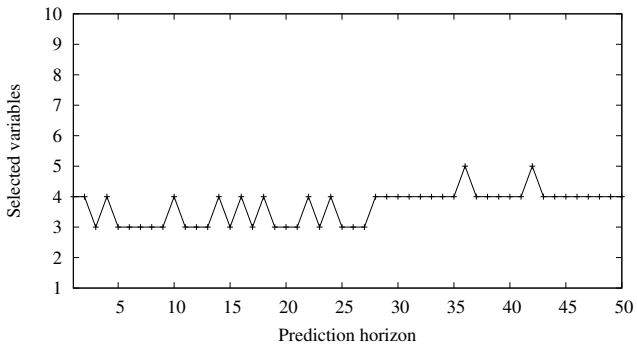
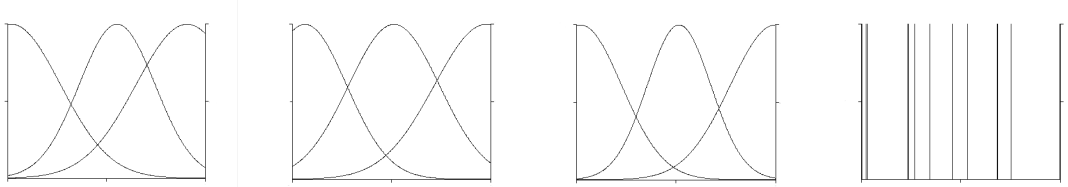



Fig. 3. ENSO: Number of selected variables for horizon up to 50. DT based selection with exhaustive search. Maximum regressor size 10.



(a) $y_{t-7}$ (b) $y_{t-2}$ (c) $y_t$ (d) $y_{t+1}$

Fig. 4. ENSO: Membership functions for the three inputs and the output of the autoregressor for prediction horizon 1, after the tuning substage. The resulting input functions are approximately the same as the initial uniform partition defined before the tuning substage.

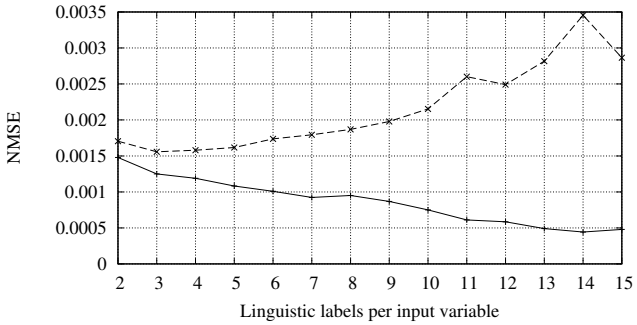



Fig. 5. ENSO: Errors for horizon 1, exhaustive DT based selection of inputs. Continuous line: training error. Dashed line: validation error.

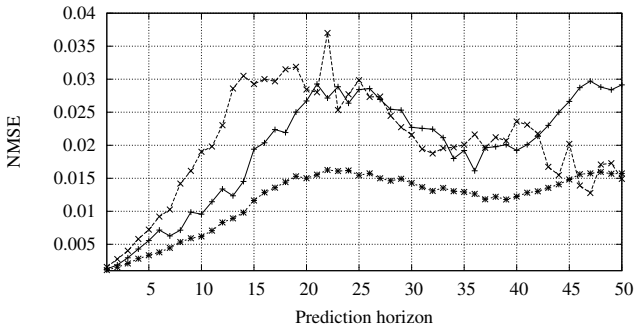



Fig. 6. ENSO: NDT estimates (∗), training (+) and validation (×) errors of fuzzy autoregressors. Maximum regressor size 10. DT based selection of inputs.

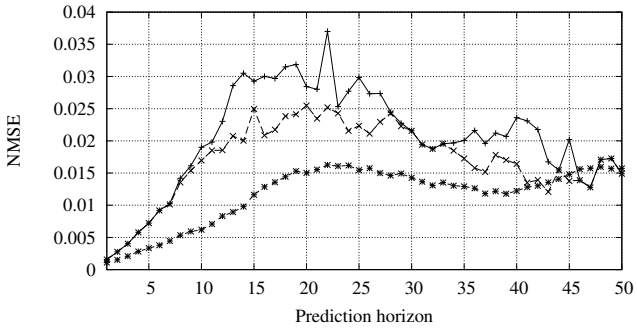



Fig. 7. ENSO: NDT estimates ($*$), test errors for the selected fuzzy autoregressors (+), validation errors for the optimal complexity selections (x). Maximum regressor size 10. DT based selection of inputs.

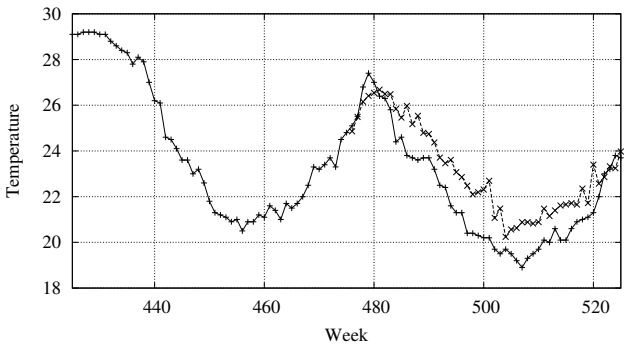



Fig. 8. ENSO: Prediction of 50 values after the training set. Continuous line (+): actual time series. Dashed line (×): predictions.

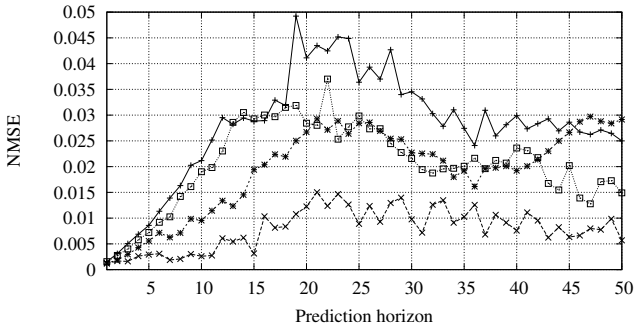
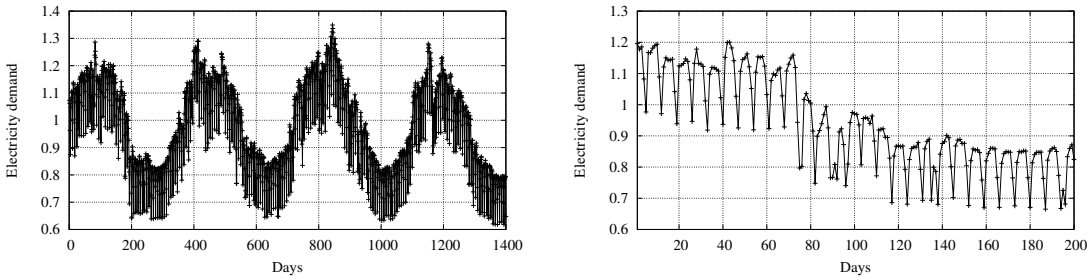



Fig. 9. ENSO: comparison of the proposed methodology against LS-SVM. Generalization errors of LS-SVM models ($+$). Generalization errors of fuzzy models ($\square$). Training errors of fuzzy models ($*$). Training errors of LS-SVM models ($\times$).




(a) Training series (1400 samples) (b) test series (201 samples).

Fig. 10. PolElec: training and test series.

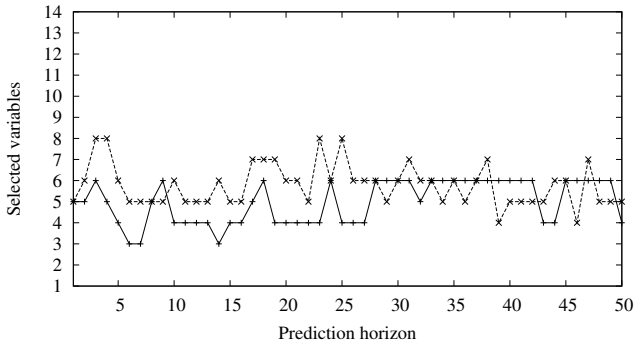
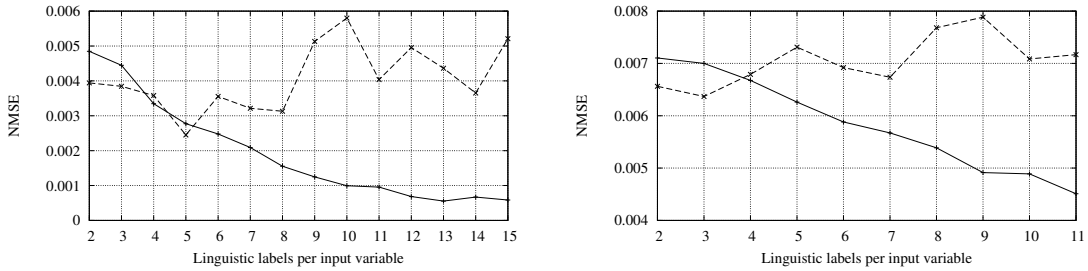
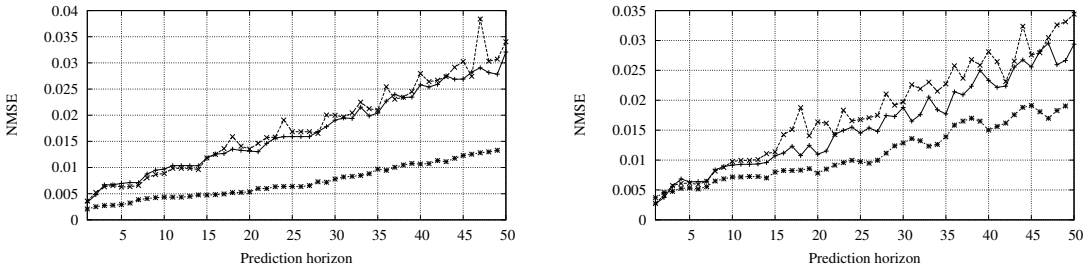



Fig. 11. PolElec: number of selected variables (exhaustive DT based selection). Continuous line: regressor size 7. Dashed line: regressor size 14.




(a) Horizon 1 (b) Horizon 7

Fig. 12. PolElec: training (continuous line) and test (dashed line) errors against linguistic labels per input. Exhaustive DT based selection of variables with maximum regressor size 7.




(a) Maximum regressor size 7 (b) Maximum regressor size 14

Fig. 13. PolElec: NDT estimates ($*$), training (+) and test ($\times$) errors of fuzzy autoregressors. Exhaustive DT based selection of inputs.

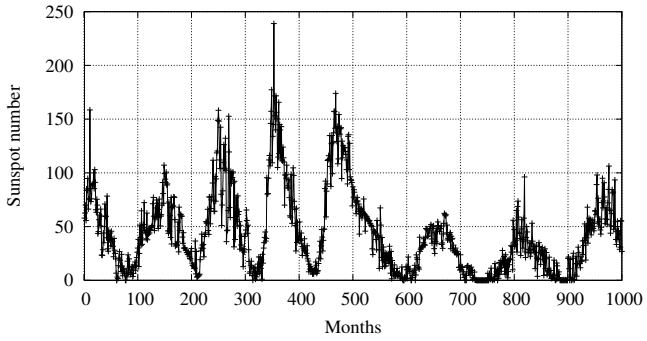



Fig. 14. Sunspots: training (first 1000 samples) and test (last 2098 samples) series.

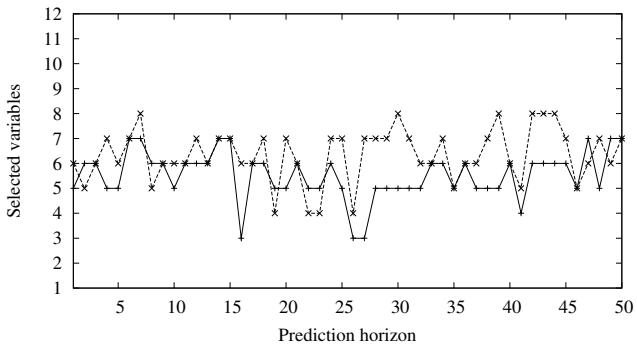
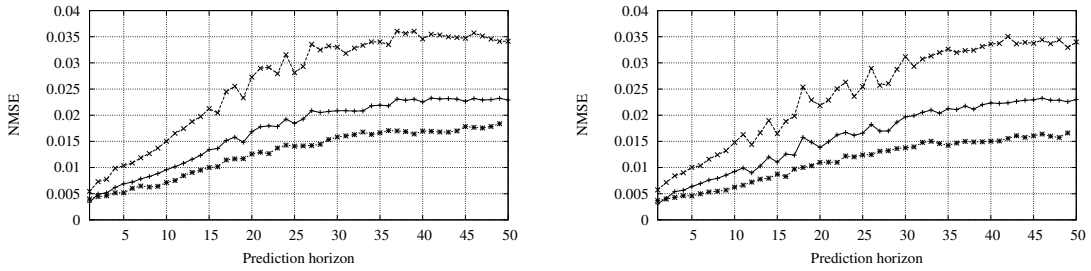



Fig. 15. Sunspots: number of selected variables (exhaustive DT based selection). Continuous line: regressor size 9. Dashed line: regressor size 12.




(a) Maximum regressor size 9

(b) Maximum regressor size 12

Fig. 16. Sunspots: NDT estimates ($*$), training (+) and test ($\times$) errors of fuzzy autoregressors. Exhaustive DT based selection of inputs.

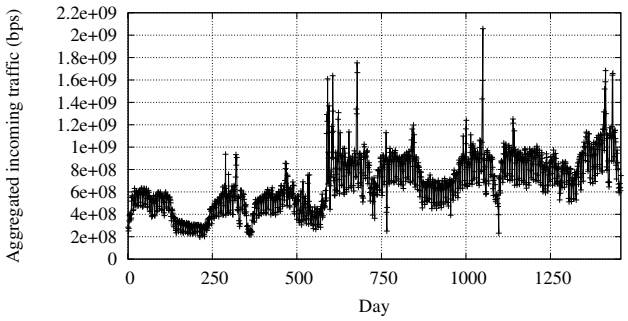



Fig. 17. Internet2: daily averaged aggregated incoming traffic in the Abilene backbone for 1458 days. Training series (first 728 values) and test series (last 730 values).

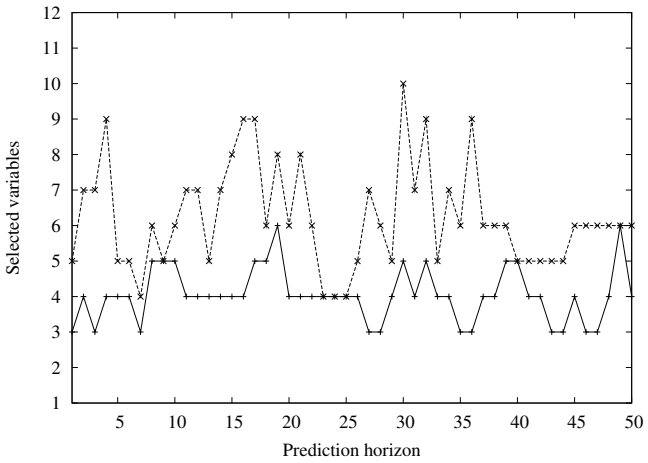
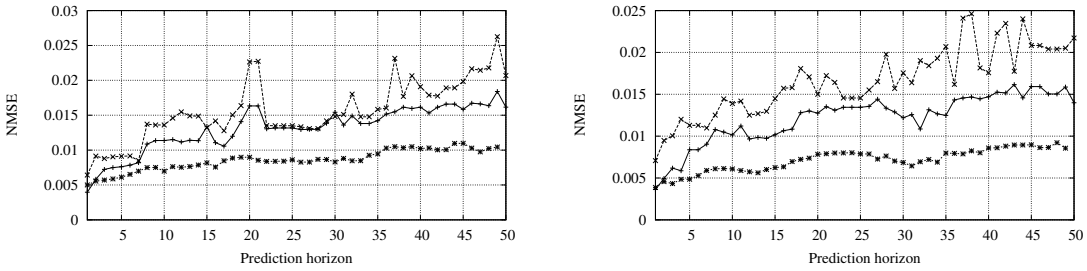
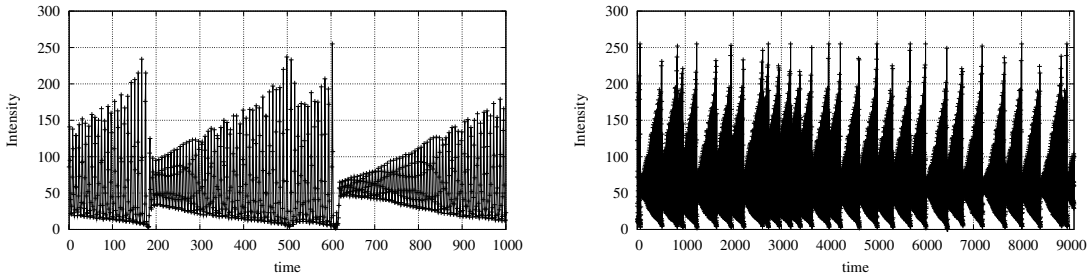



Fig. 18. Internet2: number of selected variables (exhaustive DT based selection). Continuous line: regressor size 7. Dashed line: regressor size 12.




(a) Maximum regressor size 7 (b) Maximum regressor size 12

Fig. 19. Internet2: NDT estimates ($*$), training (+) and test ($\times$) errors of fuzzy autoregressors. Exhaustive DT based selection of inputs.




(a) Training series (1000 samples) (b) Test series (9093 samples)

Fig. 20. SFL: training and test series.

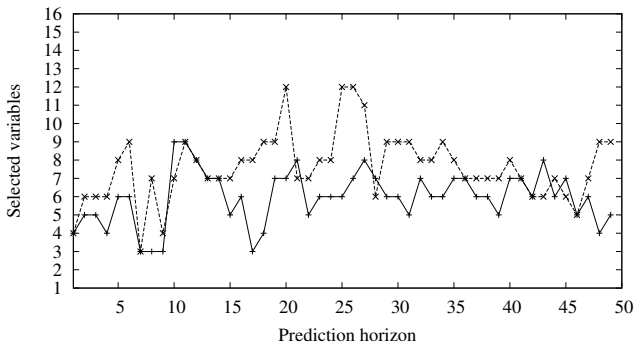
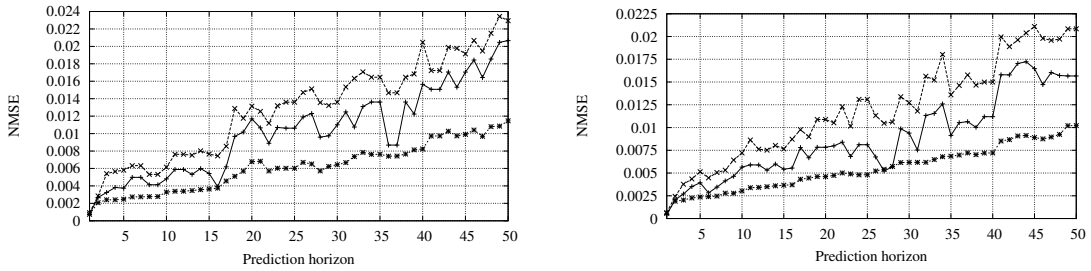



Fig. 21. SFL: Number of selected variables. Continuous line: exhaustive DT search with maximum regressor size 10. Dashed line: forward-backward DT search with maximum regressor size 16.




(a) Maximum regressor size 10 (b) Maximum regressor size 16

Fig. 22. SFL: NDT estimates (∗), training (+) and test (×) errors of fuzzy autoregressors. Exhaustive DT based selection of inputs for a maximum regressor size of 10. For regressor size 16, the exhaustive search for size 10 is extended using a forward-backward DT based search.

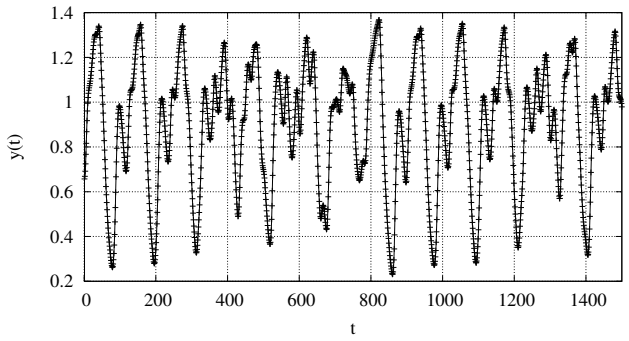



Fig. 23. MG: fragment of the Mackey-Glass series (1500 samples). The first 500 samples are selected as training set. The remaining 1000 samples are selected as test set.

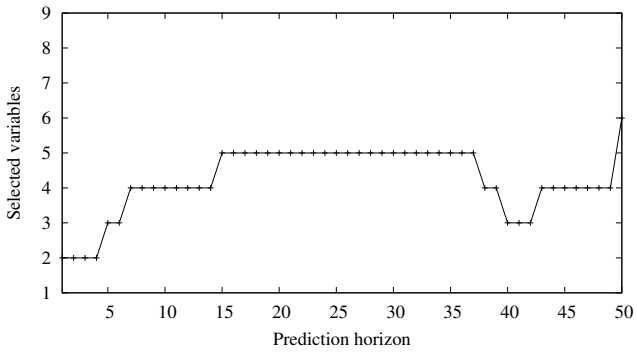



Fig. 24. MG: Number of selected variables for horizons up to 50. Exhaustive DT based selection of inputs. Maximum regressor size 9.

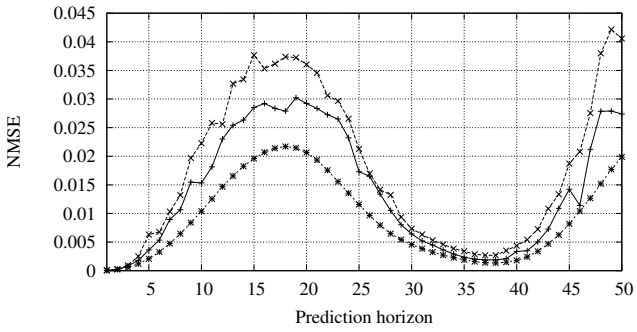



Fig. 25. MG: NDT estimates ($*$), training (+) and test ($\times$) errors of fuzzy autoregressors. Exhaustive DT based selection of inputs. Maximum regressor size 9.