Brief Papers

Time Series Modeling and Forecasting Using Memetic Algorithms for Regime-Switching Models

Christoph Bergmeir, Isaac Triguero, Daniel Molina, José Luis Aznarte, and José Manuel Benítez

***Abstract*— In this brief, we present a novel model fitting procedure for the neuro-coefficient smooth transition autoregressive model (NCSTAR), as presented by Medeiros and Veiga. The model is endowed with a statistically founded iterative building procedure and can be interpreted in terms of fuzzy rule-based systems. The interpretability of the generated models and a mathematically sound building procedure are two very important properties of forecasting models. The model fitting procedure employed by the original NCSTAR is a combination of initial parameter estimation by a grid search procedure with a traditional local search algorithm. We propose a different fitting procedure, using a memetic algorithm, in order to obtain more accurate models. An empirical evaluation of the method is performed, applying it to various real-world time series originating from three forecasting competitions. The results indicate that we can significantly enhance the accuracy of the models, making them competitive to models commonly used in the field.**

***Index Terms*— Autoregression, memetic algorithms, neuro-coefficient smooth transition autoregressive model (NCSTAR), regime-switching models, threshold autoregressive model (TAR).**

I. Introduction

Time series prediction and modeling is an important interdisciplinary field of research, involving among others Computer Sciences, Statistics, and Econometrics. Made popular by Box and Jenkins [1] in the 1970s, traditional modeling procedures combine linear autoregression (AR) and moving average. But, since data are nowadays abundantly available, often complex patterns that are not linear can be extracted. So, the need for nonlinear forecasting procedures arises. Commonly used in this context are procedures, such as multilayer perceptrons or support vector machines [2], and recent developments focus on recurrent neural networks [3], [4], generalized regression neural networks [5], and other regression procedures from machine learning.

But following the ideas of Box and Jenkins [1], a special type of nonlinear models, mainly developed by Tong [6], are piecewise linear models, which allow for modeling a series using various linear models assigned to different zones of the series. A threshold variable is then used to switch between the linear models. Many different types of these so-called threshold AR models can be found in the literature [6], [7]. If the threshold variable is chosen to be a lagged value of the time series, the model is called self-exciting threshold AR. Taking into account that the series are usually continuous, it may often be better if the change from one regime to the other is not performed by a sudden change, but merely using a smooth, differentiable function, such as the Gaussian or logistic function, which leads to smooth transition AR models.

Using this theory, Medeiros and Veiga [8] developed the neuro-coefficient smooth transition AR (NCSTAR), which uses a neural network to learn the threshold variable as a weighted sum of the inputs from the training data. Furthermore, those authors presented an iterative building procedure based on statistical testing to define the number of hidden units of the neural network.

All these models (in the following, we call the model family *TAR) have the advantage that there is a theory to translate them to fuzzy rule-based systems (FRBS) [9]–[12], which makes their interpretability more accessible.

Accuracy and interpretability are usually considered contradictory goals. In many modeling approaches, accuracy is strived for and interpretability is hardly considered. In contrast, the focus of fuzzy modeling initially was to obtain interpretable systems with acceptable accuracy, as the seminal purpose of FRBSs is to exploit the descriptive power of linguistic variables in linguistic fuzzy modeling [13]. Only in later developments, the focus was broadened to concentrate solely on accuracy in precise fuzzy modeling, and nowadays often a tradeoff between accuracy and interpretability is aimed at, e.g., by using multiobjective optimization algorithms [13].

So, although the sole use of FRBSs does not guarantee interpretability, their overall design as systems of rules makes them more accessible to humans. Furthermore, the question of measuring interpretability of FRBSs is an important subject of ongoing research in the area [14], and there exists certain consensus that important matters for interpretability are the overall number of rules, and easily understandable rule premises with few input variables [14]. Also, FRBSs are used in the literature to give interpretations to not only *TAR models [9], but also to other model classes like, e.g., neural networks [15], and support vector machines [16].

Manuscript received December 9, 2011; revised August 28, 2012; accepted August 28, 2012. Date of publication October 3, 2012; date of current version October 15, 2012. This work was supported in part by the Spanish Ministry of Science and Innovation under Project TIN-2009-14575. The work of C. Bergmeir was supported by a scholarship from the Spanish Ministry of Education of the Programa de Formación del Profesorado Universitario.

C. Bergmeir, I. Triguero, and J. M. Benítez are with the Department of Computer Science and Artificial Intelligence, CITIC-UGR, University of Granada, Granada 18071, Spain (e-mail: c.bergmeir@decsai.ugr.es; isaaktriguero@gmail.com; j.m.benitez@decsai.ugr.es).

D. Molina is with the Department of Computer Science and Engineering, University of Cadiz, Cadiz 11001, Spain (e-mail: dmolina@decsai.ugr.es).

J. L. Aznarte is with the Department of Artificial Intelligence, Universidad Nacional de Educación a Distancia, Madrid 21110, Spain (e-mail: jlaznarte@decsai.ugr.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2216898

As the iterative building procedure of the NCSTAR model controls the overall number of regimes, the resulting models typically have only few rules, and can be considered interpretable in this sense.

For model identification and estimation, the NCSTAR model uses a combination of a grid search (GS) procedure and a local search (LS) to optimize its parameters. This optimization step is crucial during the iterative building procedure, as it both influences the behavior of the test that determines the number of regimes, and the overall accuracy of the method. On the other hand, evolutionary algorithms (EAs) [17] have proven to be very efficient techniques in various fields of optimization problems [18], especially in the optimization of neural network parameters [19], [20]. They have also been applied in various time series prediction problems [21]–[24].

In particular, memetic algorithms [17], [25] are well-suited for continuous optimization, where high precision in the solutions has to be achieved [26], as they combine the evolutionary paradigm with LS strategies. In this brief, we will use memetic algorithms based on local search chains (MA-LS-Chains) [27]. Our aim is to combine the strength of EAs to find good parameters, with the benefits of the NCSTAR model, in order to develop a building procedure which results in equally interpretable, but more accurate models for time series (in comparison to the original NCSTAR method).

The structure of this brief is as follows. Section II details the theoretical background of the threshold autoregressive models. Section III discusses the memetic algorithm scheme employed, namely MA-LS-Chains. Section IV presents the proposed algorithm, which combines the NCSTAR model with the MA-LS-Chains optimization method. Section V presents the performed experimental setup, and Section VI discusses the results. Finally, Section VII concludes this brief.

II. Neuro-Coefficient Smooth Transition Autoregressive Model: NCSTAR

We define an autoregressive model for a time series $x(t)$ with a function F and a series $e(t)$ of independent, identically distributed error terms in the following way:

$$x(t) = F(x(t-1), \ldots, x(t-d)) + e(t). \tag{1}$$

The delay parameter d determines which lagged values are used as input for the method. From this general definition, various time series models can be derived according to the choice of F.

In a linear autoregression, F performs a linear combination of the past values

$$x(t) = a_0 + \sum_{i=1}^{d} a_i x(t-i) + e(t). \tag{2}$$

With $\mathbf{z}(t) = [1, x(t-1), x(t-2), \ldots, x(t-d)]^T$, and $\mathbf{a} = [a_0, \ldots, a_d]$, (2) becomes in vector notation

$$x(t) = \mathbf{a}\mathbf{z}(t) + e(t). \tag{3}$$

When F is to be a nonlinear function, a popular approach is to use mixtures of linear models

$$x(t) = \sum_{j=1}^{k} f_j(th_j(t))\mathbf{a}_j\mathbf{z}(t) + e(t). \tag{4}$$

Here, two important lines of research can be found in the literature [6], [7]: 1) regarding the (nonlinear) functions f_j that are used for mixing and 2) composition of the threshold function $th_j(t)$. This function can, for instance, take into account exogenous variables, lagged values, or combinations of both. In the threshold AR model (TAR), the functions f_j are chosen to be index functions I_j that switch between the different linear models, depending on the current threshold value $th_j(t)$, using threshold constants $c_0, \ldots, c_k$ with $-\infty = c_0 < c_1 < \cdots < c_k = \infty$, in the following way:

$$I_j(th_j(t)) = \begin{cases} 1, & \text{if } th_j(t) \in (c_{j-1}, c_j] \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

In the self-exciting TAR, for instance, the threshold variable is defined as $th_j(t) := x(t-d)$ [7]. As the index function causes abrupt changes, which might not be desirable, another possibility is to use the logistic function, that is

$$f_j(th_j(t)) = (1 + \exp(-\gamma_j(th_j(t) - c_j)))^{-1}. \tag{6}$$

This defines the logistic smooth transition autoregressive model (LSTAR) [7]. Here, the parameter c_j can still be interpreted as the threshold between two regimes, and the parameter γ_j is the slope parameter of the logistic function, which determines the smoothness of the change in the logistic function [7].

There are some other possibilities to choose the functions f_j and $th_j(t)$, to generate other models. The most relevant for our work, the NCSTAR, is a modification of the LSTAR model, with

$$th_j(t) = \omega_j \mathbf{w}(t). \tag{7}$$

Here, $\mathbf{w}(t)$ is a vector containing all variables that are relevant for the threshold, that is, lagged values and/or exogenous variables. In the following, we will use $\mathbf{w}(t) = [x(t-1), x(t-2), \ldots, x(t-d)]^T$. And ω_j is a vector of weights with $\|\omega_j\| = 1$, which has the same length as $\mathbf{w}(t)$.

The NCSTAR model has some interesting properties. Medeiros and Veiga [8] presented an iterative building procedure based on statistical tests for this type of model, and Aznarte and Benítez [9] showed that "NCSTAR models are functionally equivalent to Additive TSK FRBS with logistic membership function." Thus, with the NCSTAR, we have a model at hand with a powerful iterative building procedure that can be interpreted in terms of a fuzzy rule-based system.

The parameters of NCSTAR can be divided into linear and nonlinear ones. After having fixed the nonlinear parameters, the linear parameters can be computed in a closed-form solution. In the original version of the NCSTAR [8], a combination of GS and LS is used to determine good settings for the nonlinear parameters. The nonlinear parameters are γ_j, c_j, and ω_j for $j = 1, \ldots, k$ as in (5) and (6), with $\omega_{ij} \in [-1, 1]$, and $\|\omega_j\| = 1$. Whenever during the iterative procedure a new regime is added, starting values for this regime j are chosen in the following way [8].

1) ω_j is drawn from a uniform distribution, and normalized afterwards to ensure its norm is 1. If $\omega_{1j} < 0$, $\omega_j := -\omega_j$. This is performed M times, so that we obtain M vectors ω_j^m (with $m = 1, \ldots, M$).

2) For every ω_j^m, c_j is defined to be the median of $\omega_j^m \mathbf{x}$, with $\mathbf{x}$ being the embedded time series.
3) A grid of N values is defined to choose γ_j^n (with $n = 1, \ldots, N$) as $\gamma_j^n := n$.

Hence, a total of $N \times M$ candidate solutions for the new regime are generated. The parameter γ_j is scale-dependent, so the series are normalized before NCSTAR model building. The best parameter set, appended to the regimes yet present, is used then to initialize the LS algorithm, which improves on all values of the current model. In the original publication, the Levenberg–Marquardt algorithm [28] is used for the LS.

III. Memetic Algorithms With Local Search Chains: MA-LS-Chains

Evolutionary algorithms [17] are used nowadays successfully in a wide range of optimization problems. They evaluate a population of candidate solutions and alter and combine them to new solutions, substituting iteratively the candidate solutions by better suited variants. A central idea of EAs states that, with the use of several candidate solutions, better coverage of the search space will be achieved, and getting stuck in a particular local optimum will be avoided.

When using EAs, a tradeoff has to be made between exploring new unknown areas of the search space and exploiting already known good solutions to reach the local optimum in their neighborhood. This problem is important for continuous optimization problems, such as the one addressed in this brief, as results of high precision are required.

Memetic algorithms combine EAs with LS in order to explore more intensely the most promising areas of the search space. Instead of a generational approach for the EA, where the whole population is substituted by a new one, in this context a steady-state approach is better suited, where only single individuals are substituted. When using an elitist algorithm, it is then possible to maintain the results of the LS in the population.

In the MA-LS-Chains paradigm [27], we use a steady-state genetic algorithm [29] designed to maintain population diversity high by combining the BLX–α crossover operator [30] with a high value for its associated parameter (the default is alpha $= 0.5$), the negative assortative mating strategy [31] as its selection method, replace worst as replacement strategy, and the BGA mutation operator [32].

Another central idea of MA-LS-Chains is that, not only are the individuals stored, but also the current state of the LS for each individual. As a result, it becomes possible to interrupt the LS after a fixed number of iterations (the parameter istep of the method), and later resume it from the same state. In this way, MA-LS-Chains adapts the intensity of the LS to a solution in function of the fitness of that solution. The process of interrupting and later continuing LS is called LS chaining. In these LS chainings, the final state of the LS parameters after each LS application becomes the initial point of a subsequent LS application over the same solution, continuing the LS. In this way, MA-LS-Chains applies a higher intensity to the most promising solutions. Finally, the parameter effort controls the ratio of function evaluations used for LS over those used for the genetic algorithm.

Different LS algorithms can be used within the MA-LS-Chains paradigm. MA-CMA-Chains [27] uses the covariance matrix adaptation evolution strategy (CMA-ES) [33]. Though CMA-ES is itself an EA, it performs well in detecting and exploiting local structures. A drawback is that it does not scale well with the amount of parameters, as it employs complex mathematical operations. However, in our application this is of minor importance, as the amount of parameters is relatively low (for instance, a NCSTAR with order 4 and 10 regimes has 60 nonlinear parameters).

See [27] for a more detailed discussion of the MA-CMA-Chains algorithm.

IV. NCSTAR Fitted With MA-LS-Chains

In order to apply the MA-LS-Chains paradigm to replace the combination of GS and LS of the original NCSTAR, some adjustments are necessary.

The individuals of the population of the MA-LS-Chains algorithm consist of vectors $X = [\gamma_1, \ldots, \gamma_k, c_1, \ldots, c_k, \omega_{11}, \ldots, \omega_{d1}, \ldots, \omega_{1k}, \ldots, \omega_{dk}]$, which are realizations of the nonlinear parameters γ_j, c_j, and ω_j with $j = 1, \ldots, k$ of a NCSTAR model with k transitions (and therewith $k + 1$ regimes).

The process for model building is then the following (also shown in Algorithm 1): first, the series is tested for linearity. If the linearity assumption can be rejected, the series is assumed to be nonlinear, and the iterative building procedure is started. Otherwise, a linear model is built (a one-regime NCSTAR is a linear model). Within the iterative procedure, in the kth iteration, a $(k+1)$-regime NCSTAR is built. In every iteration, the following is executed.

1) A randomly initialized regime is added to the last iterations' solution, and this solution is added to the initial population. The rest of the population is initialized randomly. A uniform distribution constrained by the parameter domains given below is used.
2) The nonlinear parameters for all k transitions are fixed with the optimization algorithm (the linear parameters are computed for every evaluation of the fitness function).
3) The residuals of the built model are tested for linearity.
4) If the test indicates that a new regime should be added, the algorithm goes back to step (1). Otherwise, the method terminates.

It is important to properly constrain the values for γ_j and c_j. A regime that is relevant only for few training data leads to numerical problems, and unreasonable linear parameters, which may lead to very unexpected forecasts. Furthermore, if the γ_j are high, the NCSTAR deteriorates to a TAR model.

So, we restrict the nonlinear parameters in the following way.

1) For a time series x, we define the domain of the γ_j to be $\gamma_j \in [0, \gamma_0 \cdot (\max(x) - \min(x))]$, with γ_0 being a parameter of the method.
2) The thresholds c_j are constrained to lie into the $[\min(x), \max(x)]$ interval. In preliminary experiments, we also evaluated the less narrow interval $[-m_{\text{th}}, m_{\text{th}}]$,

Algorithm 1 NCSTAR-MA-LS Algorithm

```
NCSTAR ← a linear model fitted to the series
k ← 0 {the current number of transitions}
Test for linearity (H0 : γ1 = 0).
if H0 is not rejected then
  {The series is assumed to be linear.}
  return NCSTAR
else
  repeat
    {Add a new regime to NCSTAR.}
    k ← k + 1
    Build a random initial population.
    Add a new, randomly initialized regime to NCSTAR.
    Add vector X of non-linear parameters of current
    NCSTAR to the population.
    Run MA-LS-Chains using the initial population.
    Store the result in NCSTAR.
    Test for the addition of another regime (H0 : γk+1 = 0).
  until H0 is not rejected
end if
return NCSTAR
```

with $m_{\text{th}} = \sqrt{d} \cdot \max(|\min(x)|, |\max(x)|)$. But because of the numerical problems mentioned earlier, the former turned out to be a better choice for the threshold domain.

3) In order to handle the constraint $\|\omega_j\| = 1$, ω_j is encoded with n-dimensional polar coordinates [34], so that $\omega_{ij} \in [0, \pi]$ for $i = 1, \ldots, (d-1)$, and $\omega_{dj} \in [0, 2\pi]$.

V. Experimental Setup

In order to analyze the performance of the proposed method, an experimental study was carried out, which is detailed in this section. We comment on the time series data and the algorithms used, as well as on the results that were obtained.

A. Time Series Used for the Experiments

We use data from the NNGC1 and NN5 forecasting competitions,[1] and from the Santa Fe [35] competition. The high-frequency data, that is, weekly, daily, and hourly data from the NNGC1, are used. The weekly data are those related to the oil industry, such as import quantities or prices. The daily series are throughput measures of several car-tunnels, and the hourly data are arrival times of planes at airports and trains at metro stations.

The NN5 competition data are daily cash withdrawal amounts at different ATMs in the UK, measured over a period of 2 years. There is a so-called full dataset consisting of 111 series, and a reduced dataset containing 11 series. We use the reduced dataset. There are missing values present in the series, so we use one of the methods proposed for this dataset [36] to fill the gaps.

Regarding the Santa Fe data, from the six datasets, we only use data where our methods are applicable. Some of the datasets have a special focus and would require special treatment. For instance, there are series with nonuniform measurement intervals, series with missing values, and a problem where the objective is to learn a concept out of many series.

The augmented Dickey–Fuller test [37] was applied to the series, in order to use only stationary series (*TAR models are, as ARMA models, only applicable to stationary series [6]). Furthermore, series are excluded for which the linearity test, performed during the building procedure, suggests linearity. In this case, the one-regime NCSTAR that is built is a linear AR model which has no nonlinear parameters, so that the optimization procedure is not executed at all.

In total, 30 series are used, which are all available in the KEEL-dataset repository [38].[2]

For the experiments, we withhold 20 percent from the end of every series as test set, and the rest of the data is used for model building. Furthermore, as mentioned above, the series are normalized to zero mean and unitary variance. The normalization parameters are computed on the training sets, and then applied to both training set and corresponding test set.

B. Applied Algorithms

The experiments were carried out with the programming language R [39]. Our code is based on the implementations of *TAR models in the package tsDyn [40]. The MA-LS-Chains algorithm is available in the package Rmalschains [41]. Instead of the Levenberg–Marquardt algorithm, the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm was used, which is available in R by the function optim [42].

To analyze the effects of the parameter domains discussed in Section IV, we use the original method furthermore with a box-constrained version of the algorithm, the large-scale bound-constrained BFGS (L-BFGS-B) algorithm, which is available through the same function optim [42]. We call these two algorithms the NCSTAR and the NCSTAR-BOX, respectively. The version of the algorithm using MA-CMA-Chains for optimization is called NCSTAR-MA-CMA.

The original methods were used with $N = 20$, and $M = 1000$, which are the values proposed by the original authors [8]. The $N = 20$ values for each γ_j are chosen equidistant within the domain for the γ_j as defined in Section IV, using $\gamma_0 = 20$ within all of our experiments. The LS algorithm is run with a maximum of 1000 iterations. So, this yields a number of approx. 21 000 function evaluations per newly added regime in total.

To yield comparable results, the MA-CMA-Chains algorithm is used with the same amount of function evaluations. It is used with the parameter settings effort $= 0.5$, alpha $= 0.5$ (which are the default parameters of the algorithm recommended by the authors), istep $= 300$ (the method has a low sensitivity w.r.t. this parameter [27]), and a population size of 70 individuals. Although we evaluated other parameter sets in preliminary experiments, it turned out that these values are good reliable choices.

Besides the comparison within NCSTAR models, we also performed a study comparing the proposed method with other

[1] Available at http://www.neural-forecasting-competition.com.

[2] Available at http://sci2s.ugr.es/keel/timeseries.php.

TABLE I
PARAMETERS USED THROUGHOUT THE EXPERIMENTS

Algorithm	Parameters
SVR	cost = 10, gamma = 0.2, epsilon = 0.1
MLP	size = 15, decay = 0.0147, maxit = 1000
NNET	size = 9, decay = 0.1, maxit = 1000

methods commonly employed for time series forecasting. Namely, we used ϵ-support vector regression (SVR), different versions of the multilayer perceptron (MLP) trained with standard backpropagation and with the BFGS, multivariate adaptive regression splines (MARS) [43], and a linear model.

SVR is available in R through the package e1071, which implements a wrapper to LIBSVM [44]. The BFGS-trained MLP (which in the following we call NNET) is available through the nnet package [42]. The MLP with standard backpropagation is available in the package RSNNS [45] (and will be called MLP in the following). MARS is available from the package mda.

Table I shows the parameters that are used throughout the experiments (for the methods that require parameters to be set). The parameters are determined in the following way. First, a parameter grid is defined empirically for every method. Then, the parameter set performing best on a set of test series (artificial series generated from a NCSTAR process) w.r.t. the root mean squared error (RMSE) on the respective test sets is chosen.

VI. RESULTS AND DISCUSSION

For all of the 30 series, models were trained, predictions were made on the test set, and the RMSE was computed for these predictions. As both the original and the proposed NCSTAR building procedures are nondeterministic, the whole process was executed ten times, and the respective arithmetic means of the RMSE from the ten executions were used. As the series are normalized, comparing the RMSE of the different series is feasible. Fig. 1 shows box and whisker plots of the RMSE, and Table II shows averaged values over all series. The results indicate that NCSTAR-MA-CMA performs best within the compared methods, as it yields the lowest averaged RMSE.

However, this measure may be heavily influenced by outliers, and may not represent the distribution of the errors adequately. Especially in situations like ours, where the distributions are close together (see Fig. 1), this may lead to erroneous conclusions. So, in order to perform a more sophisticated evaluation of the results, we perform an analysis of the frequencies with which the methods outperform each other. Therefore, we use nonparametric statistical tests for multiple comparisons.

Concretely, we use the Friedman rank-sum test for multiple comparisons to detect statistically significant differences, and the *post-hoc* procedure of Hochberg [46] to characterize those differences [47].[3]

[3]More information can be found on the thematic web site of SCI2S about Statistical Inference in computational intelligence and data mining. Available at http://sci2s.ugr.es/sicidm.

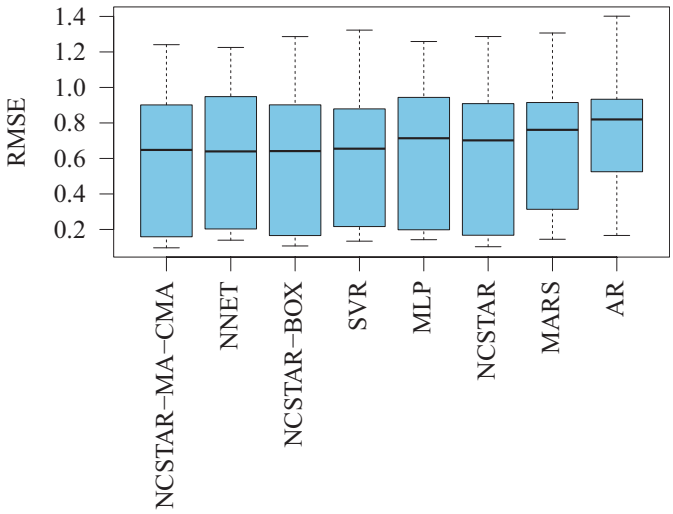

Fig. 1. Box and whisker plots for the RMSE obtained on the test sets of the series for every method. The boxes contain 50% of the data, the middle line is the median. The whiskers extend to the most extreme values.

TABLE II
RMSE AVERAGED OVER ALL SERIES, FOR THE METHODS COMPARED

	RMSE
NCSTAR-MA-CMA	0.579
NNET	0.608
NCSTAR-BOX	0.612
SVR	0.614
MLP	0.617
NCSTAR	0.623
MARS	0.667
AR	0.765

At first, we perform a comparison among the NCSTAR methods, to clearly determine the possible advantages of the method that is proposed in this brief bears. The Friedman test detects highly significant differences on a significance level of $\alpha = 0.01$ (p-value $< 2.81 \cdot 10^{-5}$). Because it obtains the best ranking, NCSTAR-MA-CMA is chosen as the control method, and the Hochberg *post-hoc* procedure is applied. Table III shows the results. As Hochberg's procedure is highly significant (with a significance level of $\alpha = 0.01$) for all compared methods, it is clear that NCSTAR-MA-CMA performs significantly better than the original versions of the algorithm.

In a second step, we compare NCSTAR-MA-CMA to the other benchmarks. The Friedman test shows highly significant differences (p-value $< 2.43 \cdot 10^{-8}$). Table IV shows the results, which indicate that NCSTAR-MA-CMA also performs significantly better than the benchmark methods we compare it to.

In combination, the results indicate that NCSTAR-MA-CMA is the best method, both in terms of the absolute value of the error as well as the frequency with which it outperforms the other methods. It especially outperforms the original algorithm, thus improving the accuracy of the generated NCSTAR models.

TABLE III
COMPARISON WITHIN THE NCSTAR METHODS. AVERAGE RANKS AND ADJUSTED p-VALUES FOR THE FRIEDMAN TEST USING THE *Post-hoc* PROCEDURE OF HOCHBERG

	Average Rank	p_{Hochberg}
NCSTAR-MA-CMA	1.37	–
NCSTAR-BOX	2.27	$4.91 \cdot 10^{-4}$
NCSTAR	2.37	$2.15 \cdot 10^{-4}$

TABLE IV
COMPARISON WITH THE BENCHMARK METHODS. AVERAGE RANKS AND ADJUSTED p-VALUES FOR THE FRIEDMAN TEST USING THE *Post-hoc* PROCEDURE OF HOCHBERG

	Average Rank	p_{Hochberg}
NCSTAR-MA-CMA	1.93	–
NNET	3.23	$7.12 \cdot 10^{-3}$
SVR	3.23	$7.12 \cdot 10^{-3}$
MLP	3.36	$7.12 \cdot 10^{-3}$
MARS	4.53	$2.94 \cdot 10^{-7}$
AR	4.70	$5.09 \cdot 10^{-8}$

TABLE V
AMOUNT OF REGIMES PRESENT IN THE MODELS, AVERAGED OVER ALL SERIES AND OVER TEN RUNS

	Amount of Regimes
NCSTAR-BOX	5.467
NCSTAR	5.513
NCSTAR-MA-CMA	5.940

Another issue is the interpretability of the model. All the NCSTAR building procedures produce the same kind of models, only the amount of regimes varies. The question is whether the NCSTAR-MA-CMA procedure yields a comparable number of rules. Table V shows the results. The original procedure generates on average 5.5 regimes, and NCSTAR-MA-CMA produces with an average of 5.9 regimes approximately 0.4 regimes more than the original methods. We consider this not to be a qualitative change in the interpretability, as an average of up to nine fuzzy rules is considered interpretable by human beings.

VII. CONCLUSION

We investigated the use of memetic algorithms within the iterative NCSTAR building procedure. In every iteration, a statistical test determines if a new regime is to be added, or if the process should terminate. If a new regime is added, the whole model is readjusted using the MA-CMA-Chains algorithm. With the combination of a building procedure that is well-founded on statistics, the possibility to interpret the model as an FRBS, and the advanced optimization procedures, we obtained a model that is flexible and robust in terms of construction, interpretability, and accuracy.

The combination of the NCSTAR model with the MA-CMA-Chains algorithm for optimization produces significantly more accurate results than the original methods.

Moreover, by using a powerful optimization algorithm, NCSTAR is also competitive with other procedures commonly employed in time series forecasting with machine learning procedures. So, NCSTAR-MA-CMA is an algorithm that can be used to build accurate and interpretable models for time series.

REFERENCES

[1] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day, 1970.

[2] S. F. Crone, M. Hibon, and K. Nikolopoulos, "Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction," *Int. J. Forecast.*, vol. 27, no. 3, pp. 635–660, Jul.–Sep. 2011.

[3] L. C. Chang, P. A. Chen, and F. J. Chang, "Reinforced two-step-ahead weight adjustment technique for online training of recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1269–1278, Aug. 2012.

[4] D. Li, M. Han, and J. Wang, "Chaotic time series prediction based on a novel robust echo state network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 787–799, May 2012.

[5] W. Yan, "Toward automatic time-series forecasting using neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1028–1039, Jul. 2012.

[6] H. Tong, *Non-Linear Time Series: A Dynamical System Approach* (Oxford Statistical Science). Oxford, U.K.: Clarendon Press, 1990.

[7] P. H. Franses and D. Van Dijk, *Nonlinear Time Series Models in Empirical Finance*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[8] M. C. Medeiros and A. Veiga, "A flexible coefficient smooth transition time series model," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 97–113, Jan. 2005.

[9] J. L. Aznarte and J. M. Benitez, "Equivalences between neural-autoregressive time series models and fuzzy systems," *IEEE Trans. Neural Netw.*, vol. 21, no. 9, pp. 1434–1444, Sep. 2010.

[10] J. L. Aznarte, M. C. Medeiros, and J. M. Benitez, "Linearity testing for fuzzy rule-based models," *Fuzzy Sets Syst.*, vol. 161, no. 13, pp. 1836–1851, Jul. 2010.

[11] J. L. Aznarte, J. M. Benitez, and J. L. Castro, "Smooth transition autoregressive models and fuzzy rule-based systems: Functional equivalence and consequences," *Fuzzy Sets Syst.*, vol. 158, no. 24, pp. 2734–2745, Dec. 2007.

[12] J. L. Aznarte, M. C. Medeiros, and J. M. Benitez, "Testing for remaining autocorrelation of the residuals in the framework of fuzzy rule-based time series modelling," *Int. J. Uncertain., Fuzziness Knowl.-Based Syst.*, vol. 18, no. 4, pp. 371–387, 2010.

[13] J. Casillas, F. Herrera, R. Pérez, M. J. del Jesus, and P. Villar, "Special issue on genetic fuzzy systems and the interpretability-accuracy trade-off," *Int. J. Approx. Reason.*, vol. 44, no. 1, pp. 1–3, 2007.

[14] M .J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Inf. Sci.*, vol. 181, no. 20, pp. 4340–4360, 2011.

[15] J. M. Benitez, J. L. Castro, and I. Requena, "Are artificial neural networks black boxes?" *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1156–1164, Sep. 1997.

[16] J. L. Castro, L. D. Flores-Hidalgo, C. J. Mantas, and J. M. Puche, "Extraction of fuzzy rules from support vector machines," *Fuzzy Sets Syst.*, vol. 158, no. 18, pp. 2057–2077, 2007.

[17] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Berlin, Germany: Springer-Verlag, 2003.

[18] I. Triguero, S. García, and F. Herrera, "IPADE: Iterative prototype adjustment for nearest neighbor classification," *IEEE Trans. Neural Netw.*, vol. 21, no. 12, pp. 1984–1990, Dec. 2010.

[19] R. Rojas, *Neural Networks: A Systematic Introduction*. New York: Springer-Verlag, 1996.

[20] C. Harpham, C. W. Dawson, and M. R. Brown, "A review of genetic algorithms applied to training radial basis function networks," *Neural Comput. Appl.*, vol. 13, no. 3, pp. 193–201, 2004.

[21] H. Du and N. Zhang, "Time series prediction using evolving radial basis function networks with new encoding scheme," *Neurocomputing*, vol. 71, nos. 7–9, pp. 1388–1400, 2008.

[22] A. F. Sheta and K. De Jong, "Time-series forecasting using GA-tuned radial basis functions," *Inf. Sci.*, vol. 133, nos. 3–4, pp. 221–228, 2001.

[23] C. G. da Silva, "Time series forecasting with a non-linear model and the scatter search meta-heuristic," *Inf. Sci.*, vol. 178, no. 16, pp. 3288–3299, 2008.

[24] V. M. Rivas, J. J. Merelo, P. A. Castillo, M. G. Arenas, and J. G. Castellano, "Evolving RBF neural networks for time-series forecasting with EvRBF," *Inf. Sci.*, vol. 165, nos. 3–4, pp. 207–220, 2004.

[25] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: Model, taxonomy, and design issues," *IEEE Trans. Evol. Comput.*, vol. 9, no. 5, pp. 474–488, Oct. 2005.

[26] H. Kita, "A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms," *Evol. Comput.*, vol. 9, no. 2, pp. 223–241, 2001.

[27] D. Molina, M. Lozano, C. García-Martínez, and F. Herrera, "Memetic algorithms for continuous optimisation based on local search chains," *Evol. Comput.*, vol. 18, no. 1, pp. 27–63, 2010.

[28] J. J. Moré, "The Levenberg–Marquardt algorithm: Implementation and theory," in *Proc. Biennial Conf. Numer. Anal.*, 1978, pp. 104-116.

[29] D. Whitley, "The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best," in *Proc. 3rd Int. Conf. Genet. Algorithms*, 1988, pp. 116–121.

[30] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithm and interval schemata," in *Proc. Found. Genet. Algorithms*, 1993, pp. 187–202.

[31] C. Fernandes and A. Rosa, "A study of non-random matching and varying population size in genetic algorithm using a royal road function," in *Proc. Congr. Evol. Comput.*, 2001, pp. 60–66.

[32] H. Mühlenbein and D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm in continuous parameter optimization," *Evol. Comput.*, vol. 1, pp. 25–49, Mar. 1993.

[33] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proc. IEEE Conf. Evol. Comput.*, May 1996, pp. 312–317.

[34] L. E. Blumenson, "A derivation of n-dimensional spherical coordinates," *Amer. Math. Monthly*, vol. 67, no. 1, pp. 63–66, Jan. 1960.

[35] A. S. Weigend and N. A. Gershenfeld, *Time Series Prediction: Forecasting the Future and Understanding the Past*. Reading, MA: Addison-Wesley, 1994.

[36] J. D. Wichard, "Forecasting the NN5 time series with hybrid models," *Int. J. Forecast.*, vol. 27, no. 3, pp. 700–707, 2011.

[37] S. E. Said and D. A. Dickey, "Testing for unit roots in autoregressive-moving average models of unknown order," *Biometrika*, vol. 71, pp. 599–607, Nov. 1984.

[38] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Logic Soft Comput.*, vol. 17, nos. 2–3, pp. 255–287, 2011.

[39] R. Development Core Team, *A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2009.

[40] A. F. Di Narzo, J. L. Aznarte, and M. Stigler. (2009). *tsDyn: Time Series Analysis Based on Dynamical Systems Theory* [Online]. Available: http://CRAN.R-Project.org/package=tsDyn

[41] C. Bergmeir, D. Molina, and J. M. Benítez. (2012). *Continuous Optimization Using Memetic Algorithms with Local Search Chains (MA-LS-Chains) in R* [Online]. Available: http://CRAN.RProject.org/package=Rmalschains

[42] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*. New York: Springer-Verlag, 2002.

[43] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Stat.*, vol. 19, pp. 1–67, Mar. 1991.

[44] C.-C. Chang and C.-J. Lin. (2001). *LIBSVM: A Library for Support Vector Machines* [Online]. Available: http://www.csie.ntu.edu.tw/_cjlin/libsvm

[45] C. Bergmeir and J. M. Benítez, "Neural networks in R using the Stuttgart neural network simulator: RSNNS," *J. Stat. Softw.*, vol. 46, no. 7, pp. 1–26, 2012.

[46] Y. Hochberg and D. Rom, "Extensions of multiple testing procedures based on Simes' test," *J. Stat. Plann. Inf.*, vol. 48, no. 2, pp. 141–152, 1995.

[47] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, 2010.