Forecasting seasonal time series with computational intelligence: on recent methods and the potential of their combinations

Martin Štěpnička[a,*], Paulo Cortez[b], Juan Peralta Donate[c], Lenka Štěpničková[a]

[a] *IRAFM, Centre of Excellence IT4Innovations, Division University of Ostrava, 30. dubna 22, 701 03 Ostrava, Czech Republic*
[b] *Centro Algoritmi, Departamento de Sistemas de Informação, Universidade do Minho, Campus Azurém, 4800-058 Guimarães, Portugal*
[c] *Computer Science Department, Carlos III University, Avda. de la Universidad, 30. 28911 Leganes, Madrid, Spain*

Abstract

Accurate time series forecasting is a key issue to support individual and organizational decision making. In this paper, we introduce novel methods for multi-step seasonal time series forecasting. All the presented methods stem from computational intelligence techniques: evolutionary artificial neural networks, support vector machines and genuine linguistic fuzzy rules. Performance of the suggested methods is experimentally justified on seasonal time series from distinct domains on three forecasting horizons. The most important contribution is the introduction of a new hybrid combination using linguistic fuzzy rules and the other computational intelligence methods. This hybrid combination presents competitive forecasts, when compared with the popular ARIMA method. Moreover, such hybrid model is more easy to interpret by decision-makers when modeling trended series.

Keywords: Time series, Computational intelligence, Neural networks, Support vector machine, Fuzzy rules, Genetic algorithm

*Corresponding author. Tel.: 420-59-709 1403.
Email addresses: Martin.Stepnicka@osu.cz (Martin Štěpnička), pcortez@dsi.uminho.pt (Paulo Cortez), jperalta@inf.uc3m.es (Juan Peralta Donate), Lenka.Stepnickova@osu.cz (Lenka Štěpničková)

1. Introduction

Forecasting the future is an important tool to support individual and organizational decision making. Time Series Forecasting (TSF) predicts the behavior of a given phenomenon based solely on the past patterns of the same event. In particular, an interesting TSF variant addresses seasonal data (e.g. monthly sales). Under such analysis, multi-step ahead prediction, i.e. forecasting several periods in advance, is highly relevant (e.g. for setting early production plans) in distinct domains, such as Agriculture, Finance, Sales and Production [41].

Computational Intelligence (CI) denotes a branch of the Artificial Intelligence field that relies on heuristic algorithms inspired in biological and natural intelligence. These CI algorithms include elements of learning and adaptation (e.g. neural networks, fuzzy rules and evolutionary computation) that facilitate intelligent behavior in complex real-world problems [25].

Although mainly statistical TSF methods (e.g. Holt-Winters exponential smoothing or ARIMA methodology) are widely used in practice [41], several computational intelligence techniques have been recently proposed for TSF as well [50]. For instance, some examples of CI applied to TSF are: Artificial Neural Networks (ANN) [18], evolutionary computation [16], Support Vector Machines (SVM) [42], immune systems [49], fuzzy techniques [5], or their combinations [36, 51].

While CI methods were successfully employed in different real-world tasks and several papers on their use in TSF were published, they became more standard in data mining applications rather than in time series, where statistical methods still dominate the market [41]. Such preference for established statistical methods is due to several factors, such as conservatism of some forecasting community members [55], but mainly due to a heritage of inferior performance of the first attempts to apply CI to TSF. Moreover, recent CI approaches to TSF often ignore very important issues such as hyperparameter selection (e.g. optimal choice of ANN topology), although it has been proved that an appropriate feature and model selection for a CI TSF model is crucial in order to provide constantly better performance [19]. Similarly, some typical arguments in favor of CI models, such as interpretability and linguistic nature of fuzzy models may seem to be a sort of an unsupported claim or even an empty cliché [9].

These observations are among the main motivations for this paper, which has a fourfold goal: 1) to provide readers with a kind of tasting of distinct

methods that may serve as an alternative to standard statistical methods and that may even outperform them; 2) to introduce how these methods may be enhanced, e.g., by using the sensitivity analysis to improve a feature selection for SVM or by a genetic algorithm to search for the optimal ANN; 3) to introduce purely new combinations of interpretable linguistic fuzzy rules with improved ANN and SVM that provide both – accurate forecasting models and easy to interpret and understand descriptions of the data generating processes; 4) and finally, to challenge prior evidence on the inferior forecasting accuracy of CI in operational forecasting [18].

Therefore, we present three novel[1] CI approaches for multi-step seasonal TSF: the *Automatic Design of Artificial Neural Networks* (ADANN), which uses genetic algorithms to evolve ANN structures; the *SVM with time lag selection based on a sensitivity analysis procedure*; and the *linguistic fuzzy approach* to the trend-cycle analysis and forecasts. The first two methods from different perspectives focus on feature and model selection process for CI methods that is often omitted [20]. The latter method focuses on the interpretability issue of fuzzy models. Moreover, we propose the very new hybrid combinations of these CI methods, such that the fuzzy approach to the trend-cycle forecasts is complemented by the earlier two approaches that forecast seasonal components. The main contribution is the presentation of these novel methods and the experimental justification of their potential. Besides the achieved high quality accuracy, such models are more easy to interpret by decision-makers when modeling trended series.

The paper is organized as follows. First, in Section 2, we introduce the used forecasting methods and principles. Next, in Section 3 we describe the seasonal datasets, introduce the forecasting accuracy metrics and finally, introduce a benchmark that serves as a comparison baseline. Then, in Section 4 we present and discuss the obtained results. Finally, we conclude the paper in Section 5.

2. Forecasting methods

Before we introduce the used forecasting methods, we briefly recall the problem. Let $\{y_t \mid t = 1, \ldots, T\} \subset \mathbb{R}$ be the past values (called *in-samples*) of

[1]All three CI methods were separately proposed in the 2010 IEEE World Congress on Computational Intelligence (WCCI), under the special session "Computational Intelligence in Forecasting".

a given time series. TSF task is to built a model that analyzes the in-samples in order to forecast the future values (so-called *out-of-samples*). Thus, the task is to determine

$$\{F_t = y_t - e_t \mid t = T+1, \ldots, T+h\} \subset \mathbb{R}, \quad h \geq 1 \tag{1}$$

where e_t denotes the forecasting error that should be minimized according to an accuracy measure (see Section 3.2) and h denotes the forecasting horizon. We assume that only in-sample data is used to build such TSF model. After fitting (also known as training) a given time series model, the last known values are fed into the model and it determines the out-of-sample. In case of $h > 1$, either the model directly outputs multi-step ahead forecasts or the out-of-samples are forecasted iteratively by using 1-ahead forecasts (and the remaining up to the $h-1$ predicted values) as inputs of the model [15].

2.1. Automatic Design of Artificial Neural Networks (ADANN)

Time series processes often exhibit temporal and spatial variability and suffer by issues of nonlinearity of physical processes, conflicting spatial and temporal scale and uncertainty in parameter estimates. ANNs are flexible models that have the capability to learn the underlying relationships between the inputs and outputs of a process, without needing the explicit knowledge of how these variables are related. We recall typical examples in market predictions [24] or in meteorological [27] and network traffic forecasting [15].

As mentioned above, finding an adequate ANN model for a particular time series is a key issue. Different studies have treated with the design of an ANN from three different points of view.

- Connection weights: values for each connection in an ANN.
- Topology: number of hidden layers, hidden nodes in each layer, etc.
- Learning rules: learning factor and momentum values.

Related to the estimation of the connection weights, it is well known that learning algorithms like backpropagation usually got stuck in a local minimum [56]. Whitley et al. [64] proposed the use of evolutionary computation to search for appropriate connection weights and avoiding the local minimum problem by means of a global search. Later, Belew et al. [7] proposed a hybrid approach carrying out a global search by a genetic algorithm and

tuning better the connection weights obtained through a backpropagation-like learning algorithm. Distinct constructive/destructive methods for the evolution of topologies of ANNs have been presented [28], but those based on evolutionary computation obtain better results [43]. At last, there are some works that try to evolve the learning rules [35].

In this paper, a novel evolving hybrid system that uses both, a genetic algorithm and the backpropagation learning, is proposed. This approach involves an evolution of the ANN topology and backpropagation learning parameter, with multiple initializations.

Normalization of the time series data has to be done as an initial step and after fitting the ANN, the inverse process is carried out. This step is important as ANN with logistic activation functions output values within the range [0,1]. Time series in-samples are transformed into a pattern set with I inputs. A single neuron is placed at the output layer and multi-step forecasts are often performed using an iterative feedback of the previous forecasts [17]. Therefore, each time series is transformed into a patterns set where each pattern consists of:

$$(N_{t-I}, \ldots, N_{t-2}, N_{t-1}) \rightarrow N_t$$

where all N_i values correspond to the normalized y_i ones. This pattern set is used to train and validate each ANN generated during the Genetic Algorithm (GA) execution. Thus, the data is split into training (with the first $X\%$ data) and validation sets (with the remaining patterns), as shown in Figure 1.

The search for the best ANN design can be performed by a GA [26] using exploitation and exploration. When using such GA, there are three crucial issues: i) the solution space and what is included into a chromosome; ii) how each solution is codified into a chromosome, i.e. encoding schema; and iii) what is the fitness function.

In this work, we opted for a multilayer perceptron as the base forecasting model, with one hidden layer and backpropagation as the learning algorithm, according to [21]. Regarding the backpropagation choice, we note that we use multiple initializations (as distinct seeds are used, see Equation 2) and also evolve its learning factor. Under such scheme, backpropagation is unlikely to fall into a local minima. Moreover, backpropagation is the most used algorithm in the TSF domain and studies presenting learning algorithms that outperform backpropagation should be viewed critically, since there is a bias to publish only algorithms that outperform the standard backpropagation [39]. Also, a majority of such papers do not report all details about training

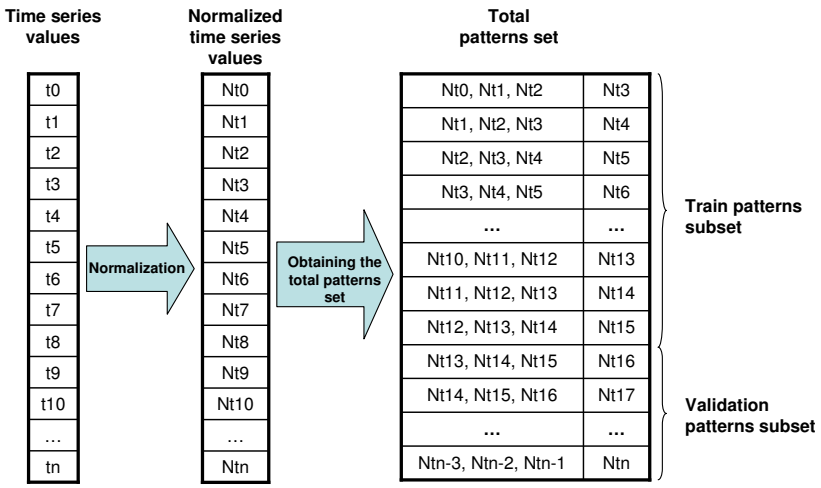

Figure 1: Process to obtain training and validation sets.

parameters and use few distinct initializations.

A direct encoding schema for fully connected multilayer perceptron is considered. For this encoding scheme the information placed into the chromosome is: two decimal digits, i.e., two genes to codify the number of inputs nodes (I); two genes for the number of hidden nodes (H); two genes for the learning factor (α); and the last ten genes for the initialization seed (s) value of the connection weights, as the seed in the Stuttgart Neural Network Simulator (SNNS) [67] is a "long int". This way, the values of I, H, α and s are obtained from the chromosome as follows:

$$\begin{aligned} \textbf{chromosome} &= g_{I_1} g_{I_2} g_{H_1} g_{H_2} g_{\alpha_1} g_{\alpha_2} g_{s_1} g_{s_2} \dots g_{s_{10}} | \forall k, g_k \in \{0, 1, \dots, 9\}, \\ s &= g_{s_1} g_{s_2} \dots g_{s_{10}}, \\ I &= 10 g_{I_1} + g_{I_2} + 1, \\ H &= 10 g_{H_1} + g_{H_2} + 1, \\ \alpha &= (10 g_{\alpha_1} + g_{\alpha_2})/100. \end{aligned} \tag{2}$$

The search process (GA) will consist of the following steps (Figure 2):

1. A randomly generated population, i.e., a set of randomly generated chromosomes, is obtained.

2. The phenotypes (ANN architectures) and fitness value of each individual of the actual generation is obtained. To obtain the phenotype associated to a chromosome and its fitness value:

   (a) The phenotype of an individual of the actual generation is first obtained (using SNNS tool).

   (b) Then for each neural network i, training and validation pattern subsets are obtained from time series data depending on the number of inputs nodes of neural network i.

   (c) The net is trained with backpropagation using SNNS [67]. When the validation error is minimal during the training process the architecture (topology and weights) is saved – early stopping. This architecture is the final phenotype of the individual.

3. The fitness is the minimum mean square validation error[2], during the learning process.

4. Once the fitness value for whole population is available the GA operators, namely elitism, selection, crossover and mutation are applied in order to generate the population of the next generation.

5. Steps 2, 3 and 4 are iteratively executed till a maximal number of generations is reached.

Since the GA works as a second order optimization procedure, the tuning of its internal parameters is not very crucial, i.e. using a population size of 46, 50 or 54 does not substantially change the results. Based on a few empirical experiments, we set the GA parameters to: population size, 50; maximum number of generations, 100; percentage of the best individual that stay unchangeable to the next generation (percentage of elitism), 10%; crossover: parents are split in one point randomly selected, offspring are the mixed of each part from parents; mutation probability will be one divided

[2]The mean square error in the fitness function is chosen in order to reduce extreme errors that may highly affect multi-step ahead forecasts. Preliminary experiments have shown that this choice leads to the best forecasts.

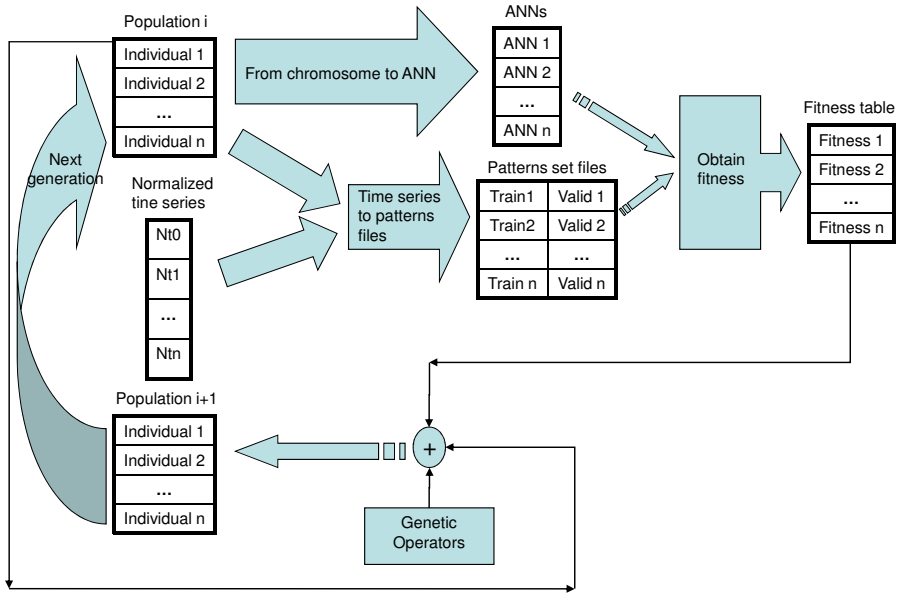

Figure 2: Schema of ANN design by GA.

by the length of the chromosome ($1/16 = 0.07$), and it will be carried out for each gene of the chromosome.

2.2. SVM - Support Vector Machine

SVM is a powerful learning tool that is based on a statistical learning theory and was developed in the 1990s due to the work of Vapnik and its collaborators [13]. It is based on two key concepts: using a kernel function SVM transforms input variables into a high dimensional feature space and then it finds the best hyperplane to model the data in the feature space.

The motivation for using SVM for forecasting is the same as for ANN: both are flexible models, i.e., with no *a priori imposed* restriction in comparison to classical TSF methods, thus both present complex nonlinear learning capabilities. However, SVMs present theoretical advantages over ANNs such

as the absence of local minima in the learning phase. Hence, SVMs have been rapidly proposed for TSF [32, 42].

When applying SVM to TSF, variable (e.g. a time lag) selection process is useful to discard irrelevant time lags in order to obtain a simpler model that is easier to interpret and that usually performs better [17, 32]. Hence, similarly to ANN, the variable selection process is a critical issue. Additionally, SVM hyperparameters such as its kernel parameter need to be adjusted [31]. Complex models may overfit the data and lose the capability to generalize, while too simple models present limited learning capabilities. We address this crucial issue by proposing a computationally efficient procedure that performs a simultaneous time lag and SVM model selection for multi-step ahead forecasting. That is the main contribution of this Section.

SVM as any regression algorithm can be applied to TSF by adopting a sliding time window of time lags $\{k_1, k_2, \ldots, k_I\}$, that is used to build a forecast. For a given time period t, the model inputs are $\mathbf{y} = (y_{t-k_I}, \ldots, y_{t-k_2}, y_{t-k_1})$ and the desired output is y_t. For example, let us consider the series 6_1,10_2,14_3, 18_4,23_5 (y_t values). If the $\{1, 3\}$ window is adopted, then two training examples can be created: $(6, 14) \rightarrow 18$ and $(10, 18) \rightarrow 23$.

In SVM regression [58], the input ($\mathbf{y}$ with domain Y) is transformed into a high m-dimensional feature space ($\Im$), by using a nonlinear mapping $\phi : Y \rightarrow \Im$ that does not need to be explicitly known but that depends on a kernel function $\kappa(x, x') =< \phi(x), \phi(x') >$, where $< u, v >$ denotes the inner product of vectors u and v. Then, the SVM algorithm finds the best linear separating hyperplane tolerating a small error ε when fitting the data in the feature space:

$$\hat{y}_{t,t-1} = w_0 + \sum_{i=1}^{m} w_i \phi(\mathbf{y}) \tag{3}$$

where $w_i \in \Re$ are coefficient weights. The ε-insensitive loss function sets an insensitive tube around the residuals and the tiny errors within the tube are discarded, see Figure 3.

We adopt the popular gaussian kernel, which presents less parameters than other kernels (e.g. polynomial) [63]: $\kappa(x, x') = exp(-\lambda||x - x'||^2)$, $\lambda > 0$. The SVM performance is affected by three parameters: λ, ε and C (a trade-off between fitting the errors and the flatness of the mapping). The kernel parameter λ produces the highest impact in the SVM performance, in comparison to C or ε. Such behavior is shown in Figure 4, which presents the sensitivity of the forecasting validation errors (in terms of boxplots) for

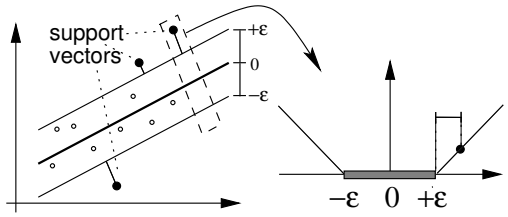

Figure 3: Example of a linear SVM regression and the ε-insensitive loss function (adapted from [58]).

the passengers series, using the time window $\{1, ..., 13\}$, when ranging each individual parameter within ten levels ($\sigma \in 2^{\{-15,...,3\}}$, $C \in 2^{\{-1,...,6\}}$ and $\varepsilon \in 2^{\{-8,-1\}}$) and fixing the remaining hyperparameter at their average values. Hence, to reduce the search space, the first two values are set using the heuristics [12]: $C = 3$ (for a standardized output) and $\varepsilon = \hat{\sigma}/\sqrt{N}$, where $\hat{\sigma} = 1.5/N \times \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ and $\hat{y}_i$ is the value predicted by a 3-nearest neighbor algorithm.

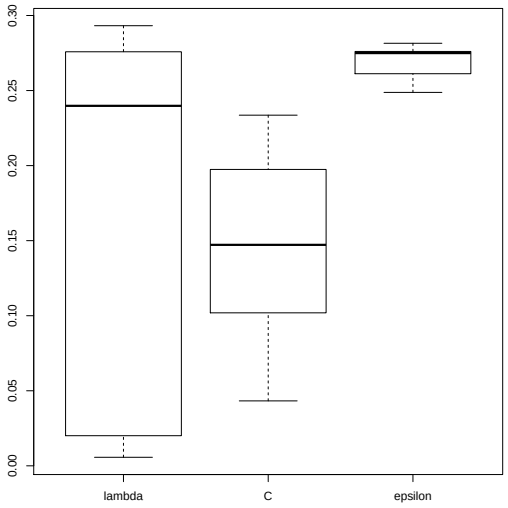

Figure 4: Example of forecasting mean absolute validation error boxplots for SVM model when λ, C and ε are changed.

Given the setup adopted, the forecasting performance is affected by both time lag and model selection. A better generalization, due to the reduced

input space, is achieved if only relevant time lags are fed into the model [32]. Also, if the kernel parameter (λ) is set with values that are too large or too small, a poor generalization will be achieved.

Sensitivity analysis [38] is a procedure that is applied after the training phase and analyzes the model responses when the inputs change. Let $\hat{y}_{t-k}(j)$ denote the output obtained by holding all input variables at their average values except y_{t-k}, which varies through its entire range with $j \in \{1, \ldots, L\}$ levels. If a given input variable y_{t-k} is relevant then it should produce a high variance V_k. Thus, its relative importance R_k can be given by:

$$\begin{array}{rcl} V_k & = & \sum_{j=1}^{L} (\hat{y}_{t-k}(j) - \overline{\hat{y}_{t-k}}(j))^2/(L-1), \\ R_k & = & V_k / \sum_{i=1}^{I} V_i \times 100\ (\%). \end{array} \quad (4)$$

This is a simple procedure that only measures single input variance and not interactions of inputs. Yet, even with this limitation, this computationally fast procedure has outperformed other more sophisticated algorithms, e.g. genetic algorithms, for the input variable selection [38].

We propose a simultaneous variable and model selection procedure for multi-step ahead forecasting. The method starts with a maximum of I_{max} time lags and iteratively deletes one input until there are no time lags. The sensitivity analysis is used to select the least relevant lag to be deleted in each iteration, allowing a reduction of the computational effort by a factor of I_{max} when compared to the standard backward selection procedure. Before feeding the SVM, all variables are standardized to a zero mean and one standard deviation. After the training, the SVM outputs are post-processed with the corresponding inverse scaling function. During a given iteration, a grid search is used to find the best model hyperparameter $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{1}\}$. The training data is divided into training and validation sets. The former, with 2/3 of the training data, is used to train the SVM model. The latter, with the remaining 1/3, is used to select the best model. Similarly to ADANN, we adopted the MSE metric for selecting such a model. After the variable and model selection phase, the final model is retrained using all training data (i.e. in-samples). The last known values are fed into the model and multi-step forecasts are built iteratively by using 1-ahead predictions as inputs [15].

The SVM experiments were conducted using the **rminer** library[14] of the R tool, which adopts the Sequential Minimal Optimization algorithm to fit the model. In this work, we set $L = 6$ [38] and also, I_{max} was set to $K + 1$ where K denotes the seasonal period (i.e., 13 for monthly time series). The

intention is to include all up to the seasonal lag plus an additional one that may be relevant for trended series.

2.3. Linguistic fuzzy approach

So far, a notable number of works aiming at fuzzy approach to time series modeling and prediction has been published. For instance, a study presenting Takagi-Sugeno rules [61] in the view of the Box-Jenkins methodology [5] or a direct fuzzification of ARIMA [62] have been published. Analogously, various neuro-fuzzy approaches that lie on the border between ANN, Takagi-Sugeno models and evolving fuzzy systems, are very often successfully used [36, 40, 57].

However, the Takagi-Sugeno rules use functional consequents without any linguistic meanings and do not employ any kind of logical implication; evolving system usually well tune (Gaussian) fuzzy sets to have a center, say, at node 5.6989 and the width parameter equal to 2.8893, see [40]. Hence, most of such fuzzy approaches, although very powerful, disregarded the importance of interpretability – leading to results that are actually black-box functions that do not provide any meaningful linguistic information [9].

Motivated by this lack of interpretability, we deal with the linguistic approach initiated in [48, 59]. There, the authors were motivated by the fact that a decomposition of a time series assumes the privilege of the interpretable model where the interpretability is meant in a sense of "readability" for non-statisticians and non-mathematicians. Even more, interpretability issue is even strengthened by using fuzzy rules of a linguistic nature to describe and forecast the trend-cycle of a given time series.

The main idea of the approach is as follows. First, a time series is decomposed into the so called *trend-cycle* [68] and the *seasonal component* using a special technique called *fuzzy transform*. Second, the trend-cycle is described by the so-called linguistic description[3] comprised from fuzzy rules. The fuzzy rules describe the data generating process autoregressively in an interpretable form. Finally, an autoregressive model of the seasonal components is used to forecast these components. Both forecasted components are composed together to obtain the time series forecast.

The *fuzzy transform* (F-transform) [52] is a special approximation technique that transforms a given continuous function defined on a real interval

[3]Alternatively we may use more common notion *fuzzy rule base*. The reason to introduce this term is published, e.g., in [45].

$[a, b] \subset \mathbb{R}$ into a simpler space of n-dimensional vectors $\mathbb{R}^n$, and then it transforms the result back.

First, *fuzzy partition* of $[a, b]$ (consisting of *basic functions*) is constructed. Usually, the *uniform fuzzy partition* is considered, i.e., n equidistant nodes $c_i \in [a, b]$ are fixed and c_0, c_{n+1} are defined as follows $a = c_0 = c_1$ and that $b = c_n = c_{n+1}$. The basic functions are fuzzy sets $A_i : [a, b] \to [0, 1]$ such that $A_i(x) > 0$ for $x \in (c_{i-1}, c_{i+1})$ and $A_i = 0$ elsewhere.

Given a fuzzy partition, the *(direct) F-transform* of $f : [a, b] \to \mathbb{R}$ is a vector $F_n[f] = [F_1, \ldots, F_n]$ with the *components of the F-transform*:

$$F_i = \frac{\int_a^b f(x)A_i(x)\ dx}{\int_a^b A_i(x)\ dx}, \qquad i = 1, \ldots, n \tag{5}$$

that determine averaged values of f above the corresponding basic functions. If the function f is not given analytically but only by a set of samples (measurements), the definite integrals in (5) are replaced by finite summation.

This is applied in case of a time series which is viewed as a discrete function $y(t)$ given at nodes $t = 1, \ldots, T$. Then an appropriate[4] fuzzy partition of the interval $[1, T]$ is constructed and the fuzzy transform determined:

$$Y_i = \frac{\sum_{t=1}^{T} y(t)A_i(t)}{\sum_{t=1}^{T} A_i(t)}, \qquad i = 1, \ldots, n. \tag{6}$$

The *inverse transform* converts the direct F-transform vector into a continuous function that approximates the original one. It is given as a linear combination of basic functions and the components of the F-transform:

$$y_{F,n}(t) = \sum_{i=1}^{n} Y_i A_i(t). \tag{7}$$

The optimality of the F-transform components according to the piecewise integral least square criterion or other properties such as noise reduction ability justifying the choice of this technique may be found in [52, 53, 60].

[4]Appropriateness is determined by the seasonality/frequency and consequently again interpretability. The F-transform components will later on appear in linguistic fuzzy rules, i.e., for the sake of readability and transparency, they should somehow mirror their averaging nature. For instance, in case of monthly time series, basic functions covering 12 (or 24) values of a time series express (bi-)annual averages of the measured value.

The role of the inverse F-transform of a given time series is to mode its trend-cycle and thus the seasonal component S_t can be obtained using:

$$S_t = y_t - y_{F,n}(t). \tag{8}$$

I order to forecast the trend-cycle, it is sufficient to forecast future F-transform components $Y_{n+1}, \ldots, Y_{n+\zeta}$, where ζ depends on the forecasting horizon h and the width of the basic functions. The trend-cycle forecast at time nodes $T+1, \ldots, T+h$ will be given as values of the inverse F-transform at these nodes, i.e., as $y_{F,(n+\zeta)}(T+1), \ldots, y_{F,(n+\zeta)}(T+h)$.

This F-transform component evolution may be analyzed and described using autoregressive fuzzy rules comprising a linguistic description. Both the components and their first- and/or second-order differences:

$$\Delta Y_i = Y_i - Y_{i-1}, \quad \Delta^2 Y_i = \Delta Y_i - \Delta Y_{i-1},$$

serve as antecedent and consequent variables of fuzzy rules. This leads to a linguistic description comprised of fuzzy rules such as the following one

$$\mathsf{IF}\ \Delta Y_{i-1} \text{ is } \mathcal{B}_{\Delta i-1}\ \mathsf{AND}\ Y_i \text{ is } \mathcal{B}_i\ \mathsf{THEN}\ \Delta Y_{i+1} \text{ is } \mathcal{C}_{\Delta i+1} \tag{9}$$

describing the autoregressive nature of the trend-cycle. Note, that the rules are automatically generated by the so called *linguistic learning* algorithm [8] implemented in the LFLC software package [23].

Symbols $\mathcal{B}_{\Delta i-1}, \mathcal{B}_i$ and $\mathcal{C}_{\Delta i+1}$ in (9) denote *evaluative linguistic expressions* (typically *very big, extremely small* or *roughly medium*), i.e., special expressions that are used to evaluate a quantity with a tolerance to uncertainty and imprecision. Their importance and the potential to model their meaning mathematically have been stressed already by L. A. Zadeh in [66].

The theory of the evaluative linguistic expressions is by far out of scope of this paper so, we recall only the main features and for details refer to [44]. The evaluative expressions always use one of the expressions of the basic trichotomy small (Sm), medium (Me), big (Bi) that can be modified by a specific adverb called *linguistic hedge*. These hedges (extremely (Ex), significantly (Si), very (Ve), more or less (ML), roughly (Ro), quite roughly (QR), very roughly (VR)) – either widen or narrow the meaning of the expressions and thus, they may be ordered according to their specificity.

The mathematical model of the meaning of evaluative expressions is based on *intensions* and *extensions* in various *contexts*. Obviously, the meaning of

"tall" when talking about skyscrapers or beetles is different. On the other hand, independently on the context (skyscrapers or bugs), the expression "tall" always denotes some objects on the right hand side of the notional set of possible values. This fact is modeled by an intension of an expression. Whenever a context, that is a triplet of real numbers $\langle v_L, v_M, v_R \rangle$ where $v_L < v_M < v_R$, is specified, we may project the intension to the extension of the expression that is modeled by a fuzzy set on $[v_L, v_R]$. This approach of modeling meaning of natural language expression fully obeys the paradigms of linguistics. For details see [44].

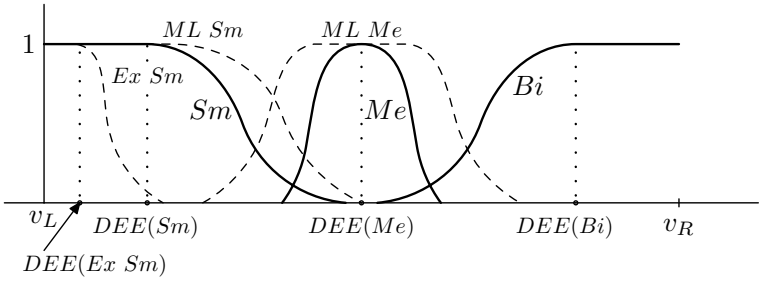

Figure 5: Fuzzy sets that model extensions of some expressions and their defuzzifications.

Using an inference, linguistic rules such as (9) are used to forecast future F-transform components. *Perception-based logical deduction* [47] is the inference that was designed for fuzzy rules with evaluative linguistic expressions. Given an input, *perception* selects the most fired rule(s), i.e., extension(s) of antecedent(s) to which the input has the highest membership degree. If there are more than one such antecedents, the most specific one(s) (according to the ordering of linguistic hedges) is/are chosen and the respective fuzzy rule(s) is/are used[5].

After the membership degree to the antecedent of a rule chosen by the perception is determined, the Łukasiewicz fuzzy implication [6] is applied

[5]This relates only to expressions with the same element of the basic trichotomy. Expressions using distinct elements of the trichotomy cannot be ordered according to their specificity.

in order to modify the respective consequent and thus to deduce a conclusion. If more rules are fired their consequences are aggregated by the minimum operation. In order to forecast the future F-transform component that is a crisp number, the *defuzzification of evaluative expressions* (DEE) is employed. Figure 5 displays defuzzified values of some typical evaluative linguistic expressions, for details see [44].

Note, that long forecasting horizons may lead to a higher number ζ of predicted F-transform components. In these situations, more steps ahead forecasting models (linguistic descriptions) may be advantageous [59].

2.4. Combination of CI techniques

While ADANN and SVM approaches introduced above aim at modeling and forecasting entire time series, the linguistic fuzzy approach focuses only on modeling and forecasting the trend-cycle of a given time series. This means that seasonal components have to be forecasted separately and composed together with the trend-cycle forecast. Seasonal components may be predicted statistically, as described in [59] or in [48], although any other TSF technique may be used as well. Given the scope of this paper, we found natural to propose the use of CI approaches, i.e., the use of ADANN and SVM, leading to two novel fuzzy hybrids, termed here Fuzzy ANN (FANN) and Fuzzy SVM (FSVM).

3. Time series data and evaluation

3.1. Time series datasets

In this work, we address seasonal data, since we believe multi-step forecasts are particularly useful for these type of series. Furthermore, seasonal series are commonly present in several domains, such as agriculture, sales, or economy. To compare the proposed TSF methods, we selected 8 benchmark time series (Table 1). Seven of them are monthly series from the well-known Hyndman's *Time Series Data Library* [33]. These are the **passengers** dataset [10] containing the information about the number (in thousands) of passengers of international airlines (Jan'49-Dec'60); **pigs** series related to numbers of pigs slaughtered in Victoria (Jan'80-Aug'95); **cars** data consisting of car sales in Quebec ('60-'68); **abraham12** represents gasoline demand at Ontario in millions of gallons ('60-'75), **milk** includes monthly milk production in pounds per cow ('62-'75), **writing** containing industry sales for printing and writing paper in thousands of French francs (Jan'63-Dec'72) and

cryer7 that collects Portland Oregon average monthly bus ridership divided by one hundred (Jan'73-Jun'82). All these seven datasets contain real-world data from different areas, which makes them interesting to forecast. First, because accurate forecasts can have an impact in their application domains. Second, these datasets suffered indirectly from external and dynamic phenomena, such as weather, economic or technological conditions that are more difficult to predict.

The last series, called **mackey-glass**, is based on the Mackey-Glass differential equation [29] and it is widely regarded as a benchmark for comparing the generalization ability of different methods. This series is a chaotic time series generated from a time-delay ordinary differential equation. This time series has been chosen in order to extend the experimental datasets by a different kind of a benchmark, i.e., by a time series that is not based on real-world data, that is not on a monthly basis, and that contains neither a trend nor a noise component.

3.2. Evaluation

The global performance of a forecasting model is evaluated by an error measure. Historically, Mean Absolute Error (MAE) or (Root) Mean Squared Error ((R)MSE) are very popular error measures. However, Mean Square Error is too sensitive to outliers [4] and furthermore, both (R)MSE and MAE are scale-dependent measures and hence, it can be hardly used for a comparison across more time series since every single time series has a different impact on the overall results [3]. For example, it has been shown that five of the 1001 series from the M-competition dominated the RMSE ranking of the forecasting methods and the remaining 996 series had only little impact on the ranking [4].

Symmetric Mean Absolute Percentage Error (SMAPE) and Mean Absolute Scaled Error (MASE) [34]:

$$\text{SMAPE} = \frac{1}{h} \sum_{t=T+1}^{T+h} \frac{|e_t|}{(|y_t| + |F_t|)/2} \times 100\%, \tag{10}$$

$$\text{MASE} = \frac{1}{h} \sum_{t=T+1}^{T+h} \frac{|e_t|}{\frac{1}{T-1}\sum_{j=2}^{T} |y_j - y_{j-1}|}, \tag{11}$$

where $e_t = y_t - F_t$ for $t = T+1, \ldots, T+h$, both belong to scale independent error measures, thus can be more easily used to compare methods across different time series.

Although the SMAPE was originally proposed in [2] in a different form, formula (10) adopts the variant used in [1] since it does not lead to negative values (ranging from 0% to 200%). However, there are still two main SMAPE drawbacks [34]: the denominator may be close to zero, and a heavier penalty is given to under-forecasting when compared to over-forecasting.

More recently proposed [34], the MASE is more widely applicable and does not hold the SMAPE disadvantages. When MASE > 1, the forecasts are worse (on average) when compared with the in-sample one-step forecasts of the naïve random-walk method. In other words MASE compares the average out-of-sample e_t with the average in-sample first difference and it relativizes the prediction error with respect to fluctuations from the past.

Generally, it is not suggested to rely only on one error measure [3, 18] since distinct results may be obtained for different measures. However, it is worth recalling the empirical evidence [18] that very good methods perform consistently well across multiple measures. For the sake of correct evaluation, we avoid choosing between SMAPE and MASE and apply both measures.

Table 1: Time series seasonal period and in-sample/out-of-sample sizes

Time Series	Seasonal Period (K)	#in-samples (T)	#out-of-samples (h_3)
passengers	12	120	24
pigs	12	164	24
cars	12	84	24
abraham12	12	168	24
milk	12	144	24
writing	12	96	24
cryer7	12	90	24
mackey-glass	30	731	60

It is worth recalling that forecasting accuracy depends upon forecasting horizons [18]. Thus, we opted to compute errors for three distinct forecasting horizons: $h_1 = K$; $h_2 = 1.5K$; and $h_3 = 2K$, where K is the seasonal period. The K, T and h_3 values for the eight benchmark series are presented in Table 1.

3.3. ARIMA by ForecastPro® as a comparison benchmark

In order to present the results with a clear insight of how good they are, a well-known method is used as a comparison benchmark. We chose the seasonal variant of the very popular seasonal Autoregressive Integrated Moving Average (ARIMA) model [10]. Note, that in order to avoid any bias from a naive implementation of ARIMA, we adopted the ForecastPro® (FP) [30] professional forecasting software. In particular, the tool was fed with the in-samples of the six datasets from Table 1 and executed the full automatic parameter selection of ARIMA to obtain the forecasts. This automatic selection includes the search for the best ARIMA variant, including its internal parameters, and detection of events such as level shifts or outliers.

The choice of FP ARIMA was straightforward because of several reasons. First, all presented CI methods are also of autoregressive nature. Second, the chosen benchmark is by far better than standard ARIMA. This is mainly due to the implementation by FP enhanced by above mentioned events detection and optimization which make the FP ARIMA a method that is difficult to outperform. This is underlined by the fact that other methods, such as exponential smoothing or mathematical curve fitting, were also tested as possible benchmarks but did not outperform FP ARIMA. Third, the automatic FP ARIMA is a popular tool that is at disposal of many forecasting professionals, who may easily check the results. Moreover, comparison to a such widely used tool has a significant explanatory value, which is fully coherent with principles of evaluating method [3]. Finally, latest advanced methods have no standardized implementations and thus, one risks that the results highly depends rather on the particular chosen implementation than on the potential of the method itself.

4. Results

4.1. Forecasting Performance

First, we analyze the performance of ADANN and SVM forecasting methods, when compared with the automatic ARIMA (FP). Accuracy was measured on all three forecasting horizons h_1, h_2, h_3 by both SMAPE and MASE (Tables 2 and 3). With respect to SMAPE, ADANN is the best option (for all horizons) for passengers, abraham12, cryer7 and partly also for writing (for h_2 and h_3). SVM outperforms the other methods for series pigs (for h_1, tie with FP for h_2), for writing (for h_1) and finally for mackey-glass series (for all horizons). Under the same metric, FP gives the best forecasts for cars

Table 2: Comparison of the individual methods (%SMAPE, best values in **bold**)

Series	**Horizon**								
	h_1			h_2			h_3		
	FP	ADANN	SVM	FP	ADANN	SVM	FP	ADANN	SVM
passengers	6.5	**2.2**	5.0	7.3	**2.5**	6.0	8.0	**2.5**	7.1
pigs	6.1	7.3	**5.8**	**6.1**	10.0	**6.1**	**7.1**	11.5	7.2
cars	**7.4**	10.6	11.0	**8.4**	9.5	10.2	**9.1**	9.8	10.0
abraham12	5.5	**4.1**	6.2	6.2	**4.5**	7.4	6.2	**3.8**	7.0
milk	**0.8**	1.4	1.1	**1.0**	1.9	1.3	**0.9**	2.3	1.3
writing	7.3	7.1	**6.3**	9.0	**7.4**	8.4	9.9	**7.7**	8.3
cryer7	9.0	**3.5**	5.6	12.1	**5.9**	7.3	13.8	**6.0**	7.4
mackey-glass	22.7	2.8	**1.4**	21.5	3.6	**1.7**	26.2	6.5	**2.4**
Mean	8.2	**4.9**	5.3	9.0	**5.7**	6.1	10.2	**6.3**	**6.3**
Median	6.9	**3.8**	5.7	7.9	**5.2**	6.7	8.6	**6.3**	7.2

and milk (all horizons) and for h_2 (jointly with SVM) and h_3 it is the best option for pigs. The comparison is very similar when adopting SMAPE. The only two differences are: ADANN is now a tie with SVM for writing series for h_1, SVM is equally best as FP for pigs and h_3.

The overall comparison is performed using the arithmetic mean and median (over all series, last rows of Tables 2 and 3) for both metrics and all horizons. When compared with the arithmetic mean, the median is more robust with respect to outliers. Globally, ADANN is the best for all series and horizons. The only exception is for $h3$, SMAPE and the arithmetic mean, where SVM is equally accurate. According both mean and median aggregation measures, FP is ranked at third place.

In Tables 4 and 5, we compare the fuzzy hybrids from Section 2.4 (FANN and FSVM) with ARIMA (FP). For both MASE and SMAPE metrics, FANN obtains the best results for passengers, abraham12 and mackey-glass. ARIMA performed generally best for pigs, cars, writing, cryer7 and milk (in this case and according to SMAPE it is tie with SVM for h_2). SVM wins only in a single case.

Overall (i.e. when considering the arithmetic mean and median), FANN is ranked at first place with respect to SMAPE for all horizons, followed by FSVM. A similar observation is found for median values of MASE errors with the only change that FSVM shares the best median with FANN for h_1. Thus we can state that although ARIMA performed best for half of the

Table 3: Comparison of the individual methods (MASE, best values in **bold**)

Series	**Horizon**								
	h_1			h_2			h_3		
	FP	ADANN	SVM	FP	ADANN	SVM	FP	ADANN	SVM
passengers	1.24	**0.42**	1.02	1.40	**0.48**	1.22	1.60	**0.51**	1.52
pigs	0.64	0.74	**0.61**	**0.64**	1.02	**0.64**	**0.76**	1.17	**0.76**
cars	**0.54**	0.70	0.72	**0.62**	0.67	0.70	**0.66**	0.68	0.69
abraham12	1.12	**0.83**	1.24	1.23	**0.89**	1.48	1.27	**0.77**	1.41
milk	**0.19**	0.31	0.25	**0.22**	0.43	0.30	**0.21**	0.53	0.29
writing	0.47	**0.42**	**0.42**	0.63	**0.50**	0.60	0.69	**0.50**	0.60
cryer7	3.06	**1.15**	1.84	4.11	**1.88**	2.39	4.77	**1.95**	2.42
mackey-glass	1.30	0.15	**0.08**	1.29	0.21	**0.10**	1.48	0.32	**0.13**
Mean	1.07	**0.59**	0.77	1.27	**0.76**	0.93	1.43	**0.80**	0.98
Median	0.88	**0.56**	0.67	0.94	**0.59**	0.68	1.02	**0.61**	0.73

series considered, its accuracy for the remaining series was not that stable, when compared with the other two methods, yielding an overall mean and median that globally ranks this method at third place. However, taking into account only the arithmetic mean of errors measured by MASE, then ARIMA outperforms both fuzzy hybrids although not significantly. More detailed discussion will be provided in Section 4.3.

4.2. Interpretability of fuzzy rules

Interpretability is often assumed to be a key feature (and advantage) of fuzzy models in various areas of application [11]. However, this aspect of fuzzy models is sometimes overused. Undoubtedly, there is a significant difference between rather numerically oriented fuzzy models such as the Takagi-Sugeno rules and models that are, say, more linguistically oriented, such as fuzzy rules with fuzzy sets that interpret both antecedents and consequents. But even in the latter case there are fundamental differences. For example, a misleading interpretation of conjunctive (Mamdani-Assilian) rules as fuzzy IF-THEN rules, although their meaning is rather different [22, 46], is a common weakness. In addition, even if the interpretation is correct, some types of treatment of the interpretations of linguistic labels with several parameters may lead to something that is very far from anything that may be called "linguistic".

Table 4: Comparison of FANN, FSVM and FP (%SMAPE, best values in **bold**)

Series	**Horizon**								
	h_1			h_2			h_3		
	FP	FANN	FSVM	FP	FANN	FSVM	FP	FANN	FSVM
passengers	6.5	**2.1**	3.0	7.3	**2.6**	3.8	8.0	**2.5**	3.9
pigs	**6.1**	6.7	7.7	**6.1**	6.7	7.9	**7.1**	8.2	7.8
cars	**7.4**	12.1	11.6	**8.4**	10.5	10.3	**9.1**	10.0	10.9
abraham12	5.5	**4.0**	4.8	6.2	**5.1**	5.5	6.2	**5.6**	5.9
milk	**0.8**	1.1	1.0	1.0	1.1	**0.9**	**0.9**	1.1	1.0
writing	**7.3**	8.8	7.5	9.0	**8.0**	9.0	9.9	**8.7**	9.9
cryer7	**9.0**	16.1	14.1	**12.1**	18.4	17.5	**13.8**	18.5	17.7
mackey-glass	22.7	**3.9**	6.8	21.5	**3.9**	10.5	26.2	**9.6**	19.0
Mean	8.2	**6.9**	7.1	9.0	**7.0**	8.2	10.2	**8.0**	9.5
Median	6.9	**5.4**	7.2	7.9	**5.9**	8.5	8.6	**8.5**	8.9

The previous sentence aims at well-tuned fuzzy models constructed with help of various tuning strategies leading to black-box functions that disregard the importance of interpretability. Let us recall the following crucial idea [9]: "*one may argue that proper input-output behavior is the central goal of automatic tuning. To some extent, this is true; however, this is not the primary mission of fuzzy systems.*" This idea perfectly addresses TSF. Even here, the accuracy of forecasts is undoubtedly the key issue. Nevertheless, we have to keep in mind the motivation behind using a fuzzy model, which generally assumed to provide an interpretable, transparent and understandable model rather than to follow only optimality goals.

We do not claim that fuzzy models should not be precise. Vice-versa, fuzzy models seem to be very promising within the forecasting area so far and any forecasting model, including a fuzzy one, should perform the TSF task with high accuracy. The goal is an interpretable model that does not necessarily "*leads to a painful loss of accuracy*" [9].

The key issue in maintaining the interpretability even in the case of a tuned fuzzy model, should be the fulfillment of several constrains on fuzzy sets that interpret linguistic expressions. Namely, they should be ordered according to natural order of linguistic expressions. That is, the interpretation of *small* should placed to the left of the interpretation of *medium* and so on. In addition, they should be convex and form a partition of the universe. Let us stress, that these constraints are fully consistent with the theory of

Table 5: Comparison of FANN, FSVM and FP (MASE, best values in **bold**)

Series	**Horizon**								
	h_1			h_2			h_3		
	FP	FANN	FSVM	FP	FANN	FSVM	FP	FANN	FSVM
passengers	1.24	**0.39**	0.56	1.40	**0.50**	0.75	1.60	**0.49**	0.80
pigs	**0.64**	0.70	0.80	**0.64**	0.71	0.81	**0.76**	0.87	0.81
cars	**0.54**	0.79	0.75	**0.62**	0.71	0.69	**0.66**	0.75	0.74
abraham12	1.12	**0.80**	0.96	1.23	**1.01**	1.09	1.27	**1.15**	1.22
milk	**0.19**	0.24	0.22	0.22	0.25	**0.21**	**0.21**	0.24	0.22
writing	**0.47**	0.61	0.52	0.63	**0.58**	0.65	0.69	**0.62**	0.69
cryer7	**3.06**	5.66	4.91	**4.11**	6.41	6.09	**4.77**	6.52	6.18
mackey-glass	1.30	**0.22**	0.36	1.29	**0.23**	0.58	1.48	**0.49**	1.03
Mean	**1.07**	1.18	1.14	**1.27**	1.30	1.36	1.43	**1.39**	1.46
Median	0.88	**0.66**	**0.66**	0.94	**0.65**	0.72	1.02	**0.69**	0.81

evaluative expressions based on the basic trichotomy of *small*, *medium* and *big* and the ordering of linguistic hedges.

A similar idea is adopted in [54] where authors claim that their tuning method does not modify the initial partition in a severe manner (and interpretability is thus kept), because the widths of membership functions change by 12.9% on average and their centers change by 3.1%. Membership functions of fuzzy sets assigned to linguistic expressions in the approach discussed in this paper do not change at all. Thus, an interpretation of each linguistic expression (its intension and given a context also its extension) is the same anywhere in any linguistic description.

To underline the interpretability and the linguistic nature of evaluative expressions and the used fuzzy IF-THEN rules, we present one of the generated models. Let us consider the **pigs** time series. In addition to the forecast itself, a user is provided by the linguistic description composed of 10 fuzzy rules symbolically displayed in Table 6. As we can see, all of the rules are purely linguistic – all the antecedents and consequents are linguistic evaluative expressions according to the respective theory. It means, that artificial fuzzy sets related to anonymous expression denoted as $\mathcal{A}_{ij}$ are not used.

Thus, every single fuzzy rule can indeed be taken as a sentence in natural language. For instance, consider the very first fuzzy rule:

IF Y_i is Bi AND ΔY_i is QR Sm AND ΔY_{i-1} is Ex Sm THEN ΔY_{i+1} is VR Sm.

Table 6: Fuzzy rules generated for the description and prediction of *pigs* time series. Abbreviations of evaluative expressions can be found in Section 2.3.

Nr.	Antecedents				Consequent
	Y_i	ΔY_i	ΔY_{i-1}	$\Rightarrow$	ΔY_{i+1}
1	Bi	QR Sm	Ex Sm	$\Rightarrow$	VR Sm
2	QR Bi	−Ro Bi	VR Sm	$\Rightarrow$	−Si Bi
3	Ex Bi	VR Sm	QR Sm	$\Rightarrow$	−Ro Bi
4	Ro Bi	Ex Sm	Sm	$\Rightarrow$	QR Sm
5	Ze	−Ex Bi	−Ro Bi	$\Rightarrow$	Ex Sm
6	Ex Sm	Ex Sm	−Ex Bi	$\Rightarrow$	Ve Sm
7	Si Sm	Ve Sm	Ex Sm	$\Rightarrow$	Sm
8	Sm	Sm	Ve Sm	$\Rightarrow$	VR Sm
9	VR Sm	VR Sm	Sm	$\Rightarrow$	VR Bi
10	QR Bi	VR Bi	VR Sm	$\Rightarrow$	−Ex Sm

It may be read as follows:

If the number of pigs slaughtered in the current year is big and the biannual increment is quite roughly small and the previous biannual increment was also positive with extremely small strength then the upcoming biannual increment will be very roughly small.

Hence, such a rule may be understood as follows. Given a big number of slaughtered pigs and with increasing and slight increasing trend from the last observation the increase will not finish but will continue with very roughly slight strength.

Similarly, we can consider the second fuzzy rule where one can find an information that having quite roughly big number of slaughtered pigs with trend that changed its direction from (very roughly) small increment to (roughly) big decrease signalizes that the trend numbers really reached a kind of saturation of the market and the number of slaughtered pigs will continue in a strong decrease.

We claim, that such readable information is an additional value that might be very helpful (e.g. to check if the model makes sense within the domain) for further decision-making and management processes. This is particularly useful for critical domain applications (e.g. control or medicine).

4.3. Discussion

When analyzing the obtained results in Section 4.1, it is clear that ADANN and SVM with the sensitivity analysis provided overall (mean, median) better results than the benchmark. Since for some time series and for some horizons the comparison benchmark was not outperformed and since we should also critically take into account the empirical nature of the comparison, we do not claim that these methods perform generally better for any time series. However, their forecasting power and potential has been clearly demonstrated.

Focusing on the precision only, ADANN seems to take an advantage in comparison to SVM equipped with the sensitivity analysis. However, also other aspects should be taken into account. Mainly the computational costs that are most preferably measured by the computation time. On a standard PC, the enhanced SVM needs up to tens of seconds (more than one hundred only for the mackey-glass time series), while ADANN requires on average tens of minutes, at some cases even above 100 minutes, and in the case of the very long mackey-glass series, the computational effort transcends more than 20 hours. In case of the monthly time series that were used for the experimental evaluation, higher time requirements are not usually a crucial problem. Nevertheless, in case of a need of an urgent decision-making or in case of a high-frequency time series (on a daily or even hourly basis) such enormous time requirements disqualify ADANN and favor SVM. For the completeness of the information, let us stress that the computational requirements for the linguistic fuzzy approach are up to few seconds and thus, do not significantly increase the requirements of FANN and FSVM in comparison to their pure ADANN and SVM versions.

The computational costs are usually closely related to another interesting aspect – model simplicity. Tables 7 and 8 show the main characteristics of the best forecasting models obtained by ADANN and SVM, respectively. Table 7 shows the number of input and hidden nodes of the topology obtained by ADANN as result for each time series. In Table 8 it can be observed the λ parameter for the kernel and which time lags are used by the sliding window. Observing the tables, we can see that sensitivity analysis search for SVM selects much simpler models, with 2 to 9 inputs, when compared with ADANN. Except for mackey-glass, SVM always includes the seasonal time lag (i.e. $K = 12$).

Similarly to the individual methods, the combined ones FANN and FSVM in overall evaluations (means and medians with respect to both metrics) outperform the benchmark with the exception – means of error measured by

Table 7: Best ADANN models.

Series	**input nodes**	**hidden nodes**
passengers	32	53
pigs	42	92
cars	32	54
abraham 12	44	111
milk	55	87
writing	37	76
cryer7	34	28
mackey-glass	15	163

Table 8: Best SVM models

Series	λ	**Window**	**#lag deletions**
passengers	2^{-7}	{1,12}	11
pigs	2^{-7}	{1,2,3,5,12,13}	7
cars	2^{-7}	{1,3,4,5,6,8,11,12,13}	4
abraham 12	2^{-9}	{1,2,10,11,12,13}	7
milk	2^{-7}	{1,12}	11
writing	2^{-5}	{12}	12
cryer7	2^{-7}	{1,12}	11
mackey-glass	2^{-3}	{1,5,6,10}	27

MASE. Observing Table 5, it is clear that the reason lies in the inaccurate predictions of FANN and FSVM in one series – cryer7. And this time series has significantly higher influence on the overall evaluation measured by MASE than the other series. And as stated above, arithmetic mean is more sensitive to such outliers when compared with the median. It is also interesting to note that measured by SMAPE, cryer7 is not that much significant in the overall evaluation. This confirms the necessity of using more that just one accuracy metric that can lead to misleading conclusions.

Since ADANN as well as SVM performed well for cryer7, it is the fuzzy approach forecasting the trend-cycle that is responsible for the weak forecasting performance of FANN and FSVM. This fact can be visually observed from

Figure 6. The problem is that there is a change in the trend-cycle development that has not been observed before and thus, can hardly be predicted. The top element of the so far nearly constantly increasing cryer7 series is only three values before the end of in-samples set. The last three decreasing values are sufficient for ADANN and SVM methods which underlines their flexibility but rather insufficient for the fuzzy approach and enhanced FP ARIMA that forecast a continually increasing trend.

In the case of the fuzzy approach, the problem is that it takes into account the components of the F-transform and these are average values. Last three decreasing values do not change the whole component sufficiently in order to provide an evidence of a decreasing trend-cycle. This is a common weakness of any method using aggregated values (recall e.g. PAA – the Piecewise Aggregate Approximation [37]) in case of an unlucky placement of the border point between the in-sample set and the out-sample set.

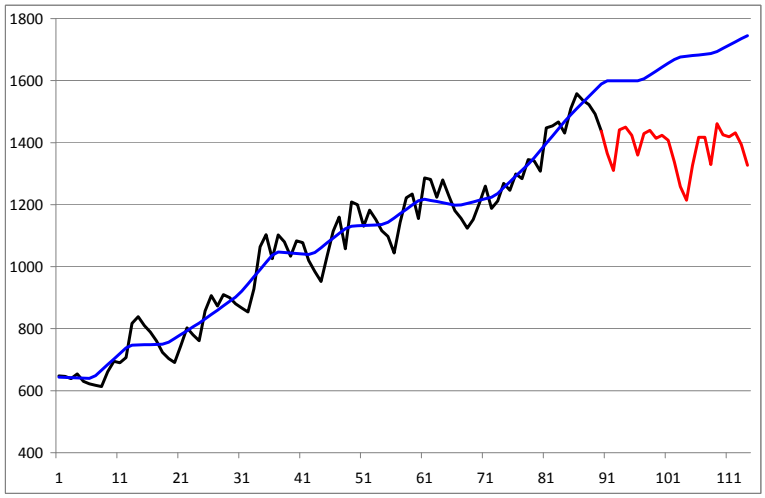

Figure 6: Graph of cryer7 time series. Black line depicts the in-samples, red line depicts the out-samples, blue line depicts the trend-cycle including its prediction.

Moreover, if we artificially delete the cryer7 time series, FANN as well as FSVM outperform not only the FP ARIMA but for some horizons also their related individual methods ADANN and SVM. So, it is a harmony

of several conditions (unobserved change in the trend-cycle development; specific placement of the border between in-samples and out-samples; too significant influence of one series to overall results with respect to a single accuracy metric) that leads to the overall evaluation that does not favor the fuzzy hybrids in comparison to the benchmark when using mean and MASE.

On the other hand, we have to stress that the proposed fuzzy approach is targeted for trended series. For comparison purposes, we applied the fuzzy variants (FANN and FSVM) to the stationary mackey-glass series, although it makes no sense to describe linguistically a trend for such series.

Table 9: Total wins when comparing ADANN and FANN (best values in **bold**)

	Horizon					
Error	h_1		h_2		h_3	
Metric	ADANN	FANN	ADANN	FANN	ADANN	FANN
SMAPE	**4**	**4**	**6**	2	**5.5**	2.5
MASE	**4**	**4**	**6**	2	**5**	3

Table 10: Total wins when comparing SVM and FSVM (best values in **bold**)

	Horizon					
Error	h_1		h_2		h_3	
Metric	SVM	FSVM	SVM	FSVM	SVM	FSVM
SMAPE	**5**	3	**5**	3	**5**	3
MASE	**5**	3	**4**	**4**	**5**	3

Tables 9 and 10 compare the individual CI methods (ADANN and SVM) with their fuzzy variants (FANN and FSVM). For the comparison, we computed the total number of pairwise wins, where the best method for a given series receives 1 point and ties count 0.5 points for both methods. While in general the individual methods (ADANN and SVM) outperform their fuzzy variants, there are horizon and error metric combinations (e.g. h_1 and $SMAPE$ for ADANN vs SVM) where the hybrid methods win in half of the series considered. Taking into account what we previously stated about

the fuzzy performance on cryer7 and mackey-glass, this is a very interesting result. In particular, if we take into account that the fuzzy models are more easy to interpret by humans, as shown in Section 4.2.

5. Conclusions

We have introduced four methods for time series forecasting from distinct Computational Intelligence (CI) subfields. The goal was fourfold: to provide readers with a kind of tasting of distinct methods that may serve as an alternative to standard statistical methods; to introduce how these methods may be improved, such as: Support Vector Machine (SVM) by the sensitivity analysis or a genetic algorithm search for an optimal Artificial Neural Network (ANN); to introduce purely new combinations of interpretable linguistic Fuzzy rules with improved ANN (FANN) and SVM (FSVM) that provide both – accurate forecasts and easy to interpret forecasting models for trended data; and finally, to challenge prior evidence on the inferior forecasting accuracy of CI in operational forecasting [18].

As a comparison baseline, we have chosen the popular seasonal ARIMA method, as implemented by the enhanced and automatic version provided by the professional ForecastPro® tool. For the comparison, we have decided to include seven monthly time series of a different nature (e.g. seasonality, trend, stationarity) and from distinct domains. Furthermore, this dataset was enriched with the well-known mackey-glass chaotic time series. Forecast accuracy was measured by two well-established and strongly motivated metrics, MASE and SMAPE, and the multi-step ahead forecasts were measured over three different horizons.

Using the particular setting (time series, forecasting horizons, error measures), the obtained results have shown that globally all proposed CI methods (i.e. ANN, SVM, FANN and FSVM) are competitive when compared with ARIMA. In general, the neural network based methods (ANN and FANN) outperformed the kernel based ones (SVM and FSVM). However, the obtained neural network models are more complex and require more computational effort (e.g. ADANN requires often several hours of computation time while SVM only demands a few minutes). Another interesting outcome is that the fuzzy CI combinations (FANN and FSVM), designed for trended seasonal series, attain a similar performance when compared with their individual CI methods. This is an interesting result, as the linguistic fuzzy

approach is more easy to interpret by humans, thus the obtained models can be more easily accepted by decision-makers.

In future research, we intend to extend the linguistic approach (FANN and FSVM) for multivariate time series that are particularly relevant in financial and economical domain where one quantity is usually explained with help of other quantities (e.g. future unemployment rate is forecasted based on past and current GDP, trade balance and retail sales). Furthermore, additional CI combinations can be explored (e.g. use of evolutionary algorithms to select the SVM time lag inputs and parameters).

Acknowledgement

The research was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070). Furthermore, we gratefully acknowledge partial support of the project KONTAKT II - LH12229 of MŠMT ČR.

[1] R. Andrawis, A. Atiya, A new Bayesian formulation for Holt's exponential smoothing, Journal of Forecasting 28(3) (2009) 218–234.

[2] J.S. Armstrong, Long-range forecasting, Wiley New York ETC., 1985.

[3] J.S. Armstrong, Evaluating methods, in: J.S. Armstrong (Eds.), Principles of Forecasting: A handbook for reasearchers and practitioners, Chap. 14, Kluwer, Boston/Dordrecht/London, 2001, pp. 443–473.

[4] J.S. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: Empirical comparisons, International Journal of Forecasting 8 (1992) 69–80.

[5] J. Aznarte, J. Benítez, J. Castro, Smooth transition autoregressive models and fuzzy rule-based systems: Functional equivalence and consequences, Fuzzy Sets and Systems 158 (2007) 2734–2745.

[6] M. Baczynski, B. Jayaram, Fuzzy Implications (Studies in Fuzziness and Soft Computing), Springer-Verlag, Heidelberg, Germany, 2008.

[7] R.K. Belew, J. Mcinerney, N. Schraudolph, Evolving Networks: Using the Genetic Algorithm with Connectionist Learning, 511–547, 1990.

[8] R. Bělohlávek, V. Novák, Learning rule base of the linguistic expert systems, Soft Computing 7 (2002) 79–88.

[9] U. Bodenhofer, P. Bauer, Interpretability of linguistic variables: a formal account, Kybernetika 2 (2005) 227–248.

[10] G. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco, 1976.

[11] J. Casillas, O. Cordón, F. H. Triguero, L. Magdalena (Eds.), Interpretability Issues in Fuzzy Modeling (Studies in Fuzziness and Soft Computing Vol. 128), Springer, Heidelberg, 2003.

[12] V. Cherkassy, Y. Ma, Practical Selection of SVM Parameters and Noise Estimation for SVM Regression, Neural Networks 17(1) (2004) 113–126.

[13] C. Cortes, V. Vapnik, Support Vector Networks, Machine Learning 20 (3) (1995) 273–297.

[14] P. Cortez, Data Mining with Neural Networks and Support Vector Machines using the R/rminer Tool, in: P. Perner (Ed.), Advances in Data Mining – Applications and Theoretical Aspects, 10th Industrial Conference on Data Mining, LNAI 6171, Springer, Berlin, Germany, 2010, pp. 572–583.

[15] P. Cortez, M. Rio, M. Rocha, P. Sousa, Internet Traffic Forecasting using Neural Networks, in: Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN 2006), IEEE, Vancouver, Canada, 2006, pp. 4942–4949.

[16] P. Cortez, M. Rocha, J. Neves, Evolving Time Series Forecasting ARMA Models, Journal of Heuristics 10 (4) (2004) 415–429.

[17] P. Cortez, M. Rocha, J. Neves, Time Series Forecasting by Evolutionary Neural Networks, Idea Group Publishing, USA, 2006, Ch. III, pp. 47–70.

[18] S.F. Crone, M. Hibon, K. Nikolopoulos, Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction, International Journal of Forecasting 27 (3) (2011) 635–660.

[19] S.F. Crone, N. Kourentzes, Feature selection for time series prediction - A combined filter and wrapper approach for neural networks, Neurocomputing 73 (2010) 1923–1936.

[20] S.F. Crone, N. Kourentzes, Naive Support Vector Regression and Multilayer Perceptron Benchmarks for the 2010 Neural Network Grand Competition (NNGC) on Time Series Prediction, in: Proceedings of 2010 IEEE Int. Joint Conf.on Neural Networks (IJCNN 2010), IEEE, Barcelona, Spain, 2010, pp. 2878–2885.

[21] G. Cybenko, Approximation by superposition of a sigmoidal function, Mathematics of Control, Signals and Systems (1989) 303–314.

[22] D. Dubois, H. Prade, What are fuzzy rules and how to use them, Fuzzy Sets and Systems 84 (1996) 169–185.

[23] A. Dvořák, H. Habiballa, V. Novák, V. Pavliska, The software package LFLC 2000 - its specificity, recent and perspective applications, Computers in Industry 51 (2003) 269–280.

[24] T. Edwards, D. S. W. Tansley, R. J. Frank, N. Davey, Northern Telecom (Nortel Limited), Traffic trends analysis using neural networks, in: Proceedings of the International Workshop on Applications of Neural Networks to Telecommuncations, 1997, pp. 157–164.

[25] A. Engelbrecht, Computational intelligence: an introduction, Wiley, 2007.

[26] D. B. Fogel, Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, 3rd Edition, IEEE Press Series on Computational Intelligence, Wiley-IEEE Press, 2005.

[27] R. J. Frank, N. Davey, S. P. Hunt, Time series prediction and neural networks, J. Intell. Robotics Syst. 31 (2001) 91–103.

[28] M. Frean, The upstart algorithm: a method for constructing and training feedforward neural networks, Neural Computation 2 (1990) 198–209.

[29] L. Glass, M. Mackey, Oscillation and chaos in physiological control systems, Science 197 (1977) 287–289.

[30] R. Goodrich, The Forecast Pro methodology, International Journal of Forecasting 16(4) (2000) 533–535.

[31] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, Springer-Verlag, NY, USA, 2008.

[32] W. He, Z. Wang, H. Jiang, Model optimizing and feature selecting for support vector regression in time series forecasting, Neurocomputing 72(1-3) (2008) 600–611.

[33] R. J. Hyndman, Time series data library, http://robjhyndman.com/TSDL/.

[34] R. J. Hyndman, A. Koehler, Another look at measures of forecast accuracy, International Journal of Forecasting 22(4) (2006) 679–688.

[35] R.A. Jacobs, Increased rates of convergence through learning rate adaptation, Neural Networks 1 (1988) 295–307.

[36] N. Kasabov, Q. Song, Denfis: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction, IEEE Transactions on Fuzzy Systems 10 (2002) 144–154.

[37] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, Knowledge and Information Systems 3 (2001) 263–286.

[38] R. Kewley, M. Embrechts, C. Breneman, Data Strip Mining for the Virtual Design of Pharmaceuticals with Neural Networks, IEEE Transactions on Neural Networks 11(3) (2000) 668–679.

[39] N. Kourentzes, S.F. Crone, Advances in forecasting with artificial neural networks, Lancaster University Management School Working Paper, The Department of Management Science, Lancaster University, 2010, URL http://eprints.lancs.ac.uk/49005/ .

[40] G. Leng, T. McGinnity, G. Prasad, An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network, Fuzzy Sets and Systems 150 (2005) 211–243.

[41] S. Makridakis, S. Wheelwright, R. Hyndman, Forecasting methods and applications, 3rd Edition, John Wiley & Sons, USA, 2008.

[42] K. Miiller, A. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, V. Vapnik, Predicting time series with support vector machines, in: Proceedings of the 7th International Conference on Artificial Neural Networks, Springer, 1997, pp. 999–1004.

[43] G.F. Miller, P.M. Todd, S.U. Hegde, Shailesh U., Designing neural networks using genetic algorithms, Proc. 3rd International Conference on Genetic Algorithms, 1989, pp. 379–384.

[44] V. Novák, A comprehensive theory of trichotomous evaluative linguistic expressions, Fuzzy Sets and Systems 159(22) (2008) 2939–2969.

[45] V. Novák, Linguistically oriented fuzzy logic controller and its design, Internat. J. Approx. Reason. 12(3-4) (1995) 263–277.

[46] V. Novák, S. Lehmke, Logical structure of fuzzy IF-THEN rules, Fuzzy Sets and Systems 157(15) (2006) 2003–2029.

[47] V. Novák, I. Perfilieva, On the semantics of perception-based fuzzy logic deduction, International Journal of Intelligent Systems 19 (2004) 1007–1031.

[48] V. Novák, M. Štěpnička, A. Dvořák, I. Perfilieva, V. Pavliska, L. Vavříčková, Analysis of seasonal time series using fuzzy approach, International Journal of General Systems 39 (2010) 305–328.

[49] I. Nunn, T. White, The application of antigenic search techniques to time series forecasting, in: GECCO, 2005, pp. 353–360.

[50] A. K. Palit, D. Popovic, Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications (Advances in Industrial Control), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[51] J. Peralta, Xiaodong Li, G. Gutierrez, A. Sanchis, Time series forecasting by evolving artificial neural networks using genetic algorithms and differential evolution, in: IJCNN, 2010.

[52] I. Perfilieva, Fuzzy transforms: theory and applications, Fuzzy Sets and Systems 157 (2006) 993–1023.

[53] I. Perfilieva, R. Valášek, Fuzzy transforms in removing noise, in: B. Reusch (Ed.), Computational Intelligence, Theory and Applications, Advances in Soft Computing, Springer, Berlin, 2005, pp. 221–230.

[54] F. M. Pouzols, A. Lendasse, A. B. Barros, Autoregressive time series prediction by means of fuzzy inference systems using nonparametric residual variance estimation, Fuzzy Sets and Systems 161 (2010) 471–497.

[55] S. Price, Mining the past to determine the future: Comments, International Journal of Forecasting 25(3) (2009) 452–455.

[56] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the RPROP algorithm, in: Proc. IEEE International Conference on Neural Networks, 1993, pp. 586–591.

[57] 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 (2006) 1260–1275.

[58] A. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 14 (2004) 199–222.

[59] M. Štěpnička, A. Dvořák, V. Pavliska, L. Vavříčková, A linguistic approach to time series modeling with the help of the F-transform, Fuzzy Sets and Systems 180 (2011) 164–184.

[60] M. Štěpnička, O. Polakovič, A neural network approach to the fuzzy transform, Fuzzy sets and Systems 160 (2009) 1037–1047.

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

[62] F. M. Tseng, G. H. Tzeng, Yu, B. J. C. Yuan, Fuzzy ARIMA model for forecasting the foreign exchange market, Fuzzy Sets and Systems 118 (2001) 9–19.

[63] W. Wang, Z. Xu, W. Lu, X. Zhang, Determination of the spread parameter in the Gaussian kernel for classification and regression, Neurocomputing 55 (3) (2003) 643–663.

[64] D. Whitley, T. Hanson, Optimizing neural networks using faster, more accurate genetic search, in: Proc. 3rd International Conference on Genetic Algorithms, 1989, pp. 391–396.

[65] L. A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.

[66] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning I, II, III, Information Sciences 8-9 (1975) 199–257, 301–357, 43–80.

[67] A. Zell, G. Mamier, R. Hübner, N. Schmalzl, T. Sommer, M. Vogt, Snns: An efficient simulator for neural nets, in: Proc. International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems (MASCOTS '93), Society for Computer Simulation International, San Diego, CA, USA, 1993, pp. 343–346.

[68] Data and Metadata Reporting and Presentation Handbook, OECD, 2005.