Naive Support Vector Regression and Multilayer Perceptron Benchmarks for the 2010 Neural Network Grand Competition (NNGC) on Time Series Prediction

Sven F. Crone, *Member, IEEE*, and Nikolaos Kourentzes

***Abstract*— In recent forecasting competitions, algorithms of Support Vector Regression (SVR) and Neural Networks (NN) have provided some of the most accurate time series predictions, but also some of the least accurate contenders failing to outperform even simple statistical benchmark methods. As both SVR and NN offer substantial degrees of freedom in model building (e.g. selecting input variables, kernel or activation functions, etc.), a myriad of heuristics and ad-hoc rules have emerged which may lead to different models with substantial differences in performance. The heterogeneity of results impairs our ability to compare the adequacy of a class of algorithms for a given dataset, and fails to develop an understanding of their presumed nonlinear and non-parametric capabilities. In order to determine a generalized estimate of performance for both SVR and NN in the absence of an accepted 'best practice' methodology, this paper seeks to compute benchmark results employing a naïve methodology which attempts to mimic many of the common mistakes in model building. The naive methodologies serve primarily as a lower error bound, representative of a within class benchmark for both algorithms in predicting the 66 time series of the NNGC Competition. In addition, their discussion aims to draw attention to the most common mistakes in modelling that regularly lead to model misspecification of MLPs and SVRs in time series forecasting.**

I. Introduction

Time series forecasting with methods of computational intelligence (CI) has received increasing attention in theory and practice. Recently, both CI-methods of Multilayer Perceptrons (MLP) and Support Vector Regression (SVR) have shown promising performance in various scientific forecasting domains [1-2], offering non-parametric, data-driven and self-adaptive approaches that learn linear or nonlinear functional relationships directly from data [3-4]. In order to objectively prove the efficacy of CI-algorithms in forecasting, outside of a controlled research experiment in which the test data is known to the researcher, their accuracy must be evaluated in a series of true ex ante comparisons against established statistical forecasting methods on empirical datasets [3, 5]. The 2010 Neural Network Grand Challenge (NNGC) provides a further opportunity to establish the forecasting accuracy of CI on six datasets of 11 empirical time series of transportation data.

Recent competitions (including the NN3, NN5 and ESTSP contests) have demonstrated that despite the presumed theoretical superiority of the model classes - founded in the promise of universal approximation for MLPs or statistical learning theory for SVR - multiple contenders from both classes have demonstrated substantial variability in their predictive accuracy, ranging from some of the top contenders of the competition to some of the most inaccurate predictions which failed to outperform even naive statistical benchmark algorithms. One potential explanation for the inhomogeneous performance of different MLPs and SVRs lies in the many degrees of freedom in modeling, requiring a data dependent selection of the meta-parameters. For example, MLPs require the setting of the number of hidden nodes and layers, choice of activation function etc. to name but a few. Similarly, SVR requires the selection of a suitable kernel function and its parameters from a set of potential functions, and two parameters to control the cost and epsilon-insensitive margin of interval scale. Both require the identification of an input vector, adequate pre- and postprocessing etc. Consequently, both MLP and SVR share common and well established challenges in model specification, offering near endless degrees of freedom in the choice of meta-parameters. As a result, a number of heuristics and ad-hoc rules-of-thumb have emerged in order to guide modeling decisions. It appears that it is this choice of combining different heuristics through an expert modeler, that determines the algorithms' performance and can result in highly accurate, or inaccurate predictions. As evidenced by the variety of approaches used in the competitions, to date no consensus exists on a valid and reliable 'best practice' methodology to specify MLPs or SVRs for forecasting. As a model class, this limits our ability to identify the efficacy of MLPs and SVRs in time series prediction in comparison to classes of statistical benchmarks, such as Exponential Smoothing or ARIMA for which best practices methodologies exist.

To establish a benchmark of forecasting accuracy for both model classes of MLPs and SVRs in time series forecasting, both are applied to forecast the 66 time series of the in the NNGC competition. While no consensus exist on a 'best practice' methodology, research has identified a number of suboptimal modeling choices which should be avoided to achieve accurate predictions. In order to determine an objective benchmark for MLP and SVR accuracy in the competition, this study seeks to compute benchmark results using a naïve methodology of a fixed parameter grid-search, which mimics common, often novice mistakes in model building. As a result, we create a lower bound of the algorithms' potential accuracy regardless of fine tuned

Manuscript received February 7, 2010. S.F. Crone and N. Kourentzes are with the Department of Management Science and the Research Centre for Forecasting, Lancaster University Management School, Lancaster LA1 4YX, United Kingdom (+44.1524.5-92991, sven.f.crone@crone.de).

978-1-4244-8126-2/10/$26.00

methodologies, to which the accuracy of individual heuristics of MLP and SVR and other algorithms may be compared.

The naïve methodology deliberately neglects relevant modeling guidelines in MLP and SVR modeling, such as input variable selection, data dependent learning rate or kernel selection, adequate scaling and preprocessing of data etc. Hence we develop a naïve benchmark as a lower bound to MLP and SVR performance. The naïve grid search estimates thousands of candidate models for 66 time series of the NNGC datasets. In addition to providing benchmark results for the NNGC competition, this universal methodology already employed in the NN3 and NN5 competition serves as a benchmark across competitions and datasets to derive generalized results of relative accuracy in comparison to other entrants.

The paper is organised as follows. First we provide a brief introduction to the NNGC competition topic and its datasets, followed by the relevant model parameters in forecasting using MLPs and ε-SVR including alternative methodologies. Section 3 outlines the naive methodologies for MLP and SVR, including input variable selection, data pre-processing, MLP and SVR parameter selection and candidate selection for the experimental setup. Due to yet undisclosed test data we provide conclusions without results of accuracy.

II. The Challenge of the NNGC Competition

The NNGC competition aims to establish the empirical accuracy of different forecasting algorithms in the domain of short term transportation and traffic forecasting, using a representative dataset of heterogeneous time series (see www.neural-forecasting-competition.com). Transportation forecasting seeks to predict the number of vehicles, travelers, internet packages, waste water etc. that will use a specific transportation facility in the future. Examples include forecasting the number of vehicles using a tunnel, the number of passengers using a subway or railway line, an airport, flight route or flight destination, or the number of ships calling on a seaport. Transportation is considered an essential prerequisite to economic prosperity, mobility and wellbeing in a civilised world, in addition to providing one of the largest service sectors worldwide. The task of the competition is to forecast a set of time series as accurately as possible, using methods from computational intelligence and applying a consistent methodology. The prediction requires regression modeling on time series data: a training set of time ordered observations is provided, representing transportation data at a specific entity, time frequency and with exogenous causal influences. The objective of the competition is to predict the next unknown realizations up to h observations into the future using a multiple step ahead forecast. The data consists of transportation data measured at different time intervals (yearly, quarterly, monthly, weekly, daily and hourly, see fig. 1 for examples), each containing 11 time series and with an individual forecasting horizon h (i.e. for yearly h=6, quarterly h=8, monthly h=12, weekly h=26, daily h=14 and hourly h=48). Contestants may choose to compete on one, two or more (up to a selection of six) datasets (we are only considering datasets of the first tournament round in 2010).

Forecasting time series of transportation demand and flows poses a number of challenges: depending on the frequency in which the time series data is sampled, it may contain a number of time series patterns including no seasonality for yearly data to multiple overlying seasonality in daily data, local time trends, structural breaks, outliers, zero and missing values etc. These are often driven by a combination of unknown and unobserved causal forces driven by the underlying yearly calendar, such as reoccurring seasonal periods, bank holidays, or special events of different length, impact, lead and lag effects.

As the task requires the prediction of a large number of time series and the data properties vary depending on its time frequency and the context of the transportation data, model selection and model parameterization specific to each time series is required for accurate prediction (e.g. if a tunnel through which transportation is flowing is located in the Swiss alps it may exhibit impacts of holiday tourism in winter, while one in a metropolitan area may have only working day calendar effects). This requires a consistent, data driven methodology for specifying a MLP or SVR, of

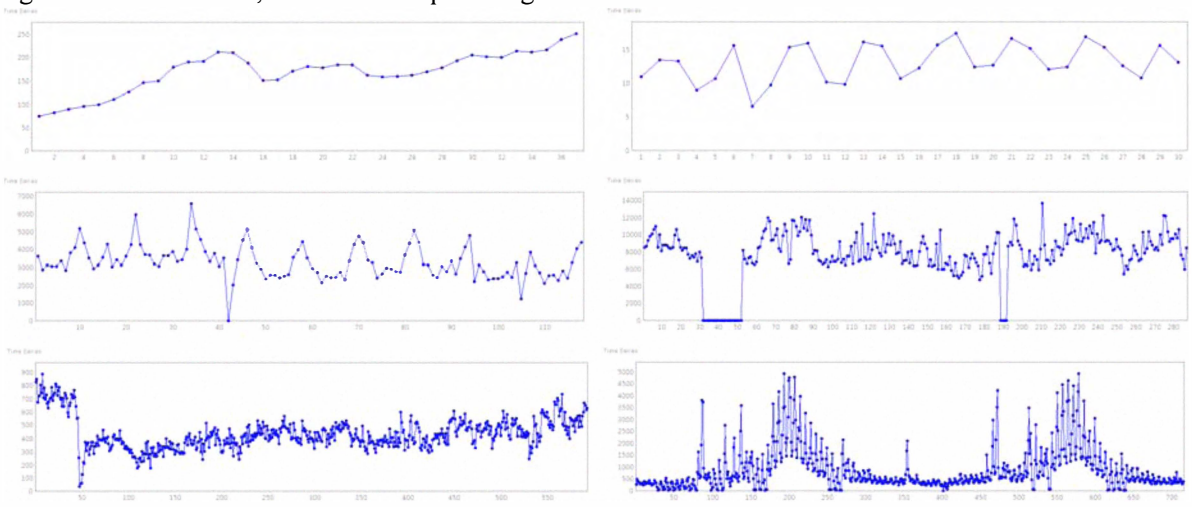

Fig.1. Examples of yearly, quarterly, monthly, weekly and daily transportation data

which some have been proposed in past competitions (with varying success). We review existing methodologies to select a simple, naive methodology eligible to serve as a lower bound on forecasting accuracy on the NNGC competition, and an objective benchmark across past competitions.

III. Forecasting Methodologies of CI

A. Modeling Computational Intelligence for Forecasting

Forecasting time series data with CI aims at constructing algorithms to map heteroassociative relationship between past time series observations and a dependent, predicted variable $\hat{y}$ of the future. The task of the algorithm is to model the underlying data generating process during training, so that a valid forecast is made when the parameterised method is subsequently presented with a new input vector value [6]. CI-algorithms such as SVR or MLPs are capable of approximating different forms of time series: using only autoregressive (AR) inputs of n lagged realisations of the dependent variable y, an algorithm can be modelled for time series forecasting, i.e. $\hat{y}_{t+1} = f(y_t, y_{t-1}, \dots, y_{t-n+1})$, or by including only explanatory variables x_i of metric or nominal scale for causal forecasting, estimating a functional relationship of $\hat{y} = f(x_1, x_2, \dots, x_z)$. An extension of these models to lagged realisations of the independent variables $x_{i,t-n}$ and the dependent variable y_{t-n} constructs more general dynamic regression, transfer function and intervention models. To extend beyond the autoregressive models of lagged realisations, the design of moving average components (MA) of past model errors in analogy to the ARIMA-Methodology of Box and Jenkins [7] enables a large range of parsimonious dynamic regression models. However, this requires recurrent architectures of MLP or SVR beyond those conventionally used. For multiple-step ahead time series prediction with both MLPs and SVRs, at a point in time t an iterative one-step ahead forecast $\hat{y}_{t+1}$ is computed using $p=n$ observations $y_t, y_{t-1}, \dots, y_{t-n+1}$ from n preceding points in time $t, t-1, t-2, \dots, t-n+1$, with n denoting the number of input variables to the algorithm.

Beyond the specification of the functional model form through the chosen CI algorithm, all algorithms require method independent choices, that equally apply to MLPs and SVRs in the next paragraphs, including the specification of the number and time lagged realisations of input variables as a rolling input vector window of fixed size over a time series, and adequate data preprocessing.

B. Forecasting with Multilayer Perceptrons

Forecasting time series with NNs is conventionally based on modelling a feed-forward topology in analogy to an non-linear autoregressive AR(p) model using the established Multilayer Perceptron (MLP) [3], to which we will limit our analysis here. As MLPs provide many degrees of freedom in determining the model form and input variables, we provide a short overview of specifying MLPs for time series modelling; an introduction is in [8].

The functional form of a single layered MLPs is characterised by its input vector $Y = [y_t, y_{t-1}, \dots, y_{t-n+1}]$, which

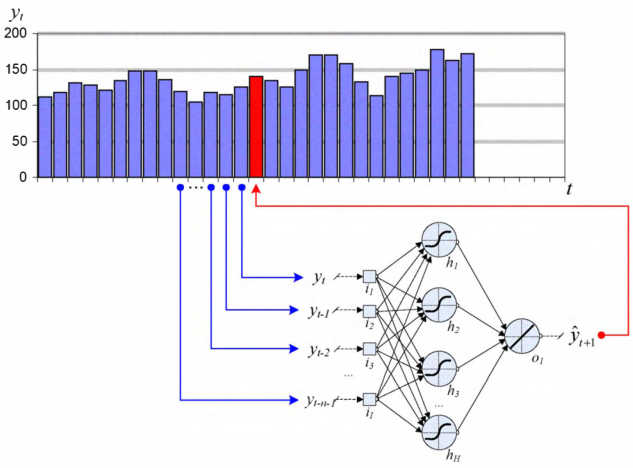

Fig. 2: Autoregressive MLP for time series forecasting

captures the lagged observations of the time series in input nodes I, the number of hidden nodes H and output nodes O in the network, and the non-linear transfer functions $g(\cdot)$ used in the nodes of the hidden layer, conventionally using the sigmoid logistic (Log) or hyperbolic tangent (TanH) functions [3]. The network parameters are denoted as weights w_{ij} connecting input, hidden and output layer respectively, and the biases w_{0j} of each neuron. The architecture and topology of a MLP is displayed in figure 1.

For parameterisation, data is presented to the MLP as an overlapping set of input vectors formed as a sliding window over the time series observations. Consequently, the specification of the network architecture determines the time series components that may be captured in the AR(p)-lags of the input vector and the capability of approximation. To specify these meta-parameters for forecasting, a variety of trial-and-error approaches, simple heuristic rules-of--thumb and more objective, replicable methodologies have been proposed, with different validity and reliability and the resulting implications for forecasting accuracy.

To date, the majority of publications employ a trial-and-error-approach based on a modeller's educated guesses and intuitive judgments which are fundamentally irreplicable, and are rarely validated beyond the dataset of a publication. In order to overcome these limitations, a multitude of simple heuristic rules have been proposed, derived from a modeller's subjective experience [9], which often provide conflicting guidance on architecture specification. In specifying the number of hidden layers, the majority of researchers limit MLP architectures to a single hidden layer (see e.g. [10], based upon an interpretation of the proof of universal approximation), while others suggest to use multiple layers [11]. Similar disagreement persists in specifying the number of hidden nodes H for a single hidden layer, resulting in a myriad of conflicting heuristics, e.g. based on the number of time series observations n, or based on the number of input nodes I, including $H=I/2$, I [12], $1.5\,I$ to $3\,I$ [10], or $2\,I+1$ [3]. (As the underlying input vectors I were pre-determined using dissimilar rules as well, these heuristics permit no interpretation nor comparison.) Similarly, no consensus exists on the use of the logistic [6,

12] or hyperbolic tangent activation functions [11] in hidden nodes, nor on linear functions in output nodes, etc. As the amount of competing rules almost matches the degrees of freedom in finding the architectural meta-parameters, this questions their help in solving the original problem and reveals the lack of consensus on how to specify MLP architectures. Only limited empirical evidence exists that the proposed heuristics resolve the problem of architecture specification [13], in particular in comparison to established statistical benchmark algorithms on larger datasets, rendering most heuristics of limited value. To guide the specification of MLP for forecasting, a number of methodologies have been proposed as a coherent collection of a set of procedures to specify NNs depending on the underlying data conditions, both for modeling generic data [9, 14] or for specific data properties of financial data [15], telecommunication data [9] etc. (for an introductory discussion see [3]). While these methodologies draw freely upon a mixture of theory, heuristics, statistical hypothesis testing procedures and algorithms they propose a consistent procedural structure to NN modelling [16]. However, most MLPs submitted to prior competitions such as NN3 or NN5 used completely distinct architectures and displayed varying accuracy, which allowed no inference on what architectural choices were beneficial, and made a coherent evaluation of the capabilities of the model class of MLPs impossible. Consequently, we seek to establish a very basic, Naive methodology (beyond mere trial and error) using heuristic rules to specify a lower bound to MLP accuracy and create a data dependent, universal benchmark across datasets and competitions.

C. Forecasting with Support Vector Regression

Also an algorithm of CI, the method of Support Vector Regression (SVR), based on statistical learning theory by Vapnik [17], estimates a linear or nonlinear function $f(\mathbf{x})$ that minimizes the forecasting error on a training data set $((\mathbf{x}_1, y_1),...,(\mathbf{x}_\ell, y_\ell)) \subseteq (\mathbf{X} \times Y)^\ell$ while keeping the functional form as flat as possible [18]. SVR is formulated as convex optimization problem with slack variables ξ_i,ξ_i^* to allow for model errors and to control the trade-off between overfitting and model complexity through a regularization parameter $C > 0$ [17]. We consider the special case of an ε-SVR, using an ε-insensitive loss function, that assigns an error only to those observations $\xi_i,\xi_i^* \geq 0$ outside an epsilon-insensitive tube of width ε [19], which are called support vectors. To handle non-linear functional relationships in forecasting problems, data is mapped from a low dimensional input to a higher dimensional feature space F using a kernel function ϕ, where the problem may be solved by exact optimisation [20]. For an introduction to SVR see [4].

In forecasting with SVR, the input vector contains the lag structure of the time series, which results in dot products after combining them with the support vectors in the kernel function. The quadratic optimization problem is solved to determine Lagrangian multipliers α_i, α_i^* that specify the SVR's parameters as weights $\upsilon_i = \alpha_i - \alpha_i^*$ [4]. The dot products are then weighted by $\upsilon_i = \alpha_i - \alpha_i^*$ to calculate the one-step-ahead prediction output together with the threshold *b* [4]. The forecasting process can be visualised as in fig. 2.

To apply ε-SVR to time series forecasting, a number of method specific meta-parameters must be determined a priori by determining the costs *C*, the width of the epsilon-insensitive loss function *ε*, the kernel function and its kernel parameters γ [21], which have a substantial impact on the forecasting accuracy of the algorithm [22]. The regularisation parameter *C* determines the trade-off between the model capacity, reflected in the flatness of the approximated function, and the amount to which deviations

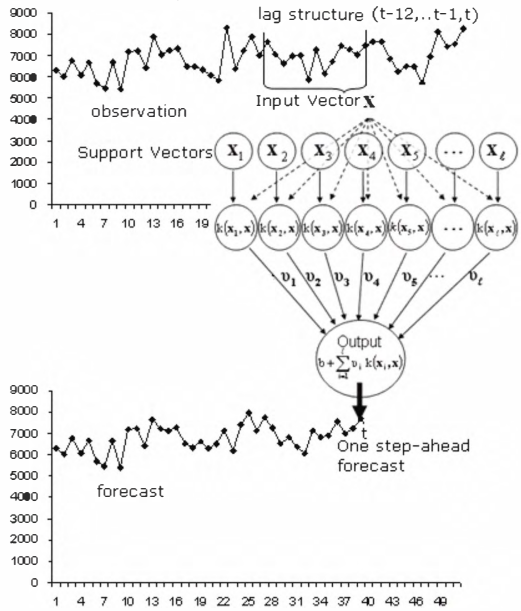

Fig. 3. Time series prediction with SVR

larger then the *ε*-insensitive tube are tolerated [19]. A larger value for *C* reduces the error contribution but yields a more complex forecasting function that is more likely to overfit on the training data [21]. Hence, it appears reasonable to evaluate parameters of C between a very small lower bound to create SVR-models with simple, flat functions to handle strong noise and a large upper bound to also consider SVR-models that describe more complex time series structures. Therefore, ε-SVR requires an expert to determine model parameters a priori in dependence of the data properties.

The ε-parameter controls the size of the ε-insensitive tube and consequently the number of support vectors and the error contributions of observations lying outside it [18]. As ε corresponds to the level of noise in a time series, large values of ε allow an approximation of the structure of the underlying functional relationship of a time series with high noise as opposed to overfitting to the noise. Prior publications use margin values of $e^{-8} \leq \varepsilon \leq e^{-1}$ [23]. In order to determine the *ε*-SVR parameters various modelling heuristics exist. Some approaches such as [24] and [25] determine the parameter ε as a linear dependency on the noise of the training data, which however assumes a priori

knowledge of the noise level [23] and hence yields little empirical benefit. The kernel parameter γ defines the width of the kernel to reflect the range of the training data in feature space and therefore the ability of an SVR to adapt to the data [26], using $e^{-8} \leq \gamma \leq e^{8}$ [23] or $2^{-8} \leq \gamma \leq 2^{1}$.

A common approach to determine suitable parameters per time series follows a systematic, step-wise grid search over the parameter space [30]. As the evaluation of every possible parameter combination would be intractable for parameters of interval scale, a grid using equidistant steps in the parameter space limits the computational effort at the cost of missing accurate parameter combinations and assuming a continuous error surface across meta-parameter choices. Different grids are applicable and will lead to different results, using linear step sizes, exponential [30] or logarithmic increasing sequences [23]. See fig. 2 for an example grid with exponentially growing sequences.

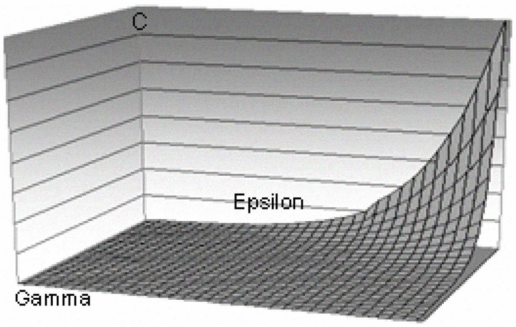

Fig. 4. A grid with exponentially growing sequences

Although previous publications employed parameter bounds of grid searches of $e^{-8} \leq C \leq e^{8}$ [30] and $2^{-2} \leq C \leq 2^{12}$ [31], they did so without theoretical justification, making the parameter bounds a further meta-parameter themselves. In addition, stepwise refinements of the grid size in parameter space are feasible, leading to an analytically simple yet computationally expensive parameter selection approach.

Valuable insight into developing robust methodologies has been generated by estimating leave-one-out bounds for SVR parameters [23] and analysing the interdependency of parameters, such as the impact of C for a given ε which has only negligible effects on the generalization performance (as long as C is larger than a threshold determined from the training data) [21], although these simplifications to limit the number of relevant parameters have not been universally received. Similarly, advanced approaches to parameter estimation, such as Bayesian frameworks for Gaussian SVR [27] and Bayesian model adaptation [28], may yield superior results to simpler heuristics of parameter grid search [29], but these still dominate most applications.

IV. Naïve Methodologies for SVR and MLP

A. Input variable selection

The forecasting accuracy of any method, including CI methods of SVR and MLP depends largely on providing adequate input information to learn from. In time series forecasting this takes the form of specifying significant timed lags of the dependent variable y_{t-n} and excluding irrelevant ones, hence determining the length of the input vector. Multiple methods exist to specify input vectors, based on simple heuristic rules such as using the full seasonal length [14], statistical autocorrelation analysis on the order of autoregressive (AR), integrated (I) and moving average (MA) processes or mixed ARIMA-processes of lagged realisations of the dependent variable [3] or using spectral analysis to detect multiple overlying seasonal patterns.

In order to mimic a naïve modelling approach we select a simple heuristic decision rule based on the observation interval of the time series, using a constant lag structure of one calendar year seasonality s, e.g. the past $s = 12$ observations for monthly, $s = 52$ weeks for weekly, or $s = 365$ days for daily data etc. in order to account for possible annual or shorter seasonality, as suggested by Balkin and Ord [14] as part of their automatic MLP model building. The lag structure was adapted to the seasonality of each dataset and used identically for all 11 time series in each dataset, despite the possibility of requiring different lag structures for different time series, suboptimal inputs due to non-parsimonious and redundant input lags, the necessity to include more, e.g. s+1 lags for seasonal integrated autoregressive processes SARIMA$(p,d,0)(P,D,0)_s$ or even longer memory for the approximation of MA processes of SARIMA$(0,0,q)(0,0,Q)_s$ by extending the input vector to multiples of the yearly seasonality. Considering recent research on input vector specification, this approach will lead to overspecified input vectors, which include a number of irrelevant variables that superimpose noise on the training data rather than identifying only the relevant ones, e.g. via stepwise regression [32]. For high frequency data in particular, this results in long input vectors that extend learning time and impair accuracy through the unnecessarily inflated degrees of freedom from superfluous weights.

B. Data pre-processing

Each of the six datasets contains 11 time series of yearly, quarterly, monthly, weekly and daily data from the NNGC competition. The time series across and within a dataset are heterogeneous, show various non-stationary seasonal and non-seasonal patterns and noise levels, including missing values, outliers, level shifts, for time series of varying length.

In data preprocessing, missing values and outliers are routinely corrected as they impair correct parameterisation: missing values are misinterpreted as valid observations of zero value by for statistical algorithms as well as MLPs and SVR. In addition, outliers in the form of single occurrences which are not representative of the normal behaviour of the data generating process, receive increased weight during parameterisation due to their large error contribution from squared error objective functions, focussing the attention of

the learning algorithm towards abnormal observations and away from the regular behaviour of the time series.

In addition to outlier removal, all data should be scaled prior to training in order to facilitate learning, speed up the computation process and to avoid numerical difficulties. While other forms of scaling are often employed, e.g. statistical normalisation using mean and standard-deviation [3], this distorts nonstationary patterns so substantially that it creates additional difficulties for the algorithm to learn, which are beyond even a naive methodology. In order not to bias results too naively, each time series observation y_t is linearly scaled as z_t into the conventional operating interval of a MLP of [0.0; 1.0], using the scaling function of

$$z_t = \frac{\left(y_t - y_{t\min}\right)}{\left(y_{t\max} - y_{t\min}\right)} \quad , \quad (3)$$

on the minimum $y_{t\,min}$ and maximum value $y_{t\,max}$ of y_t on both training and validation set. This linear scaling does not allow for instationary time series, such as trends continuing their increase into the unseen test data, which will exceed the representation bounds of the algorithm, e.g through possible saturation effects of the nodes of a MLP. Although this shortcoming can easily be overcome by rescaling into a smaller interval, e.g. applying a headroom of 50% to avoid saturation effects, it is often omitted in experiments, further biasing naive CI-predictions.

C. Naive Multilayer Perceptrons

After pre-specifying the number of input nodes and data preprocessing ranges for the MLP, we employ an exhaustive approach of a grid search of the number of hidden nodes from none to 20 in steps of 2, n^H=[0, 1, ..., 20] in only a single hidden layer. In combination with the number of input nodes this creates MLP topologies from only a few to multiple hundreds of parameters and hence degrees of freedom, often exceeding the number of observations in the time series and creating potential to overfitting to the data. Each node uses a logistic transfer function, although evidence from experiments has shown faster convergence and improved results from a Hyperbolic Tangent [11].

For training, we initialise each MLP five times with random starting weights uniformly distributed in [-1, 1], a common number of initialisations for network training although research has demonstrated that substantially more initialisations are required to find robust minima in even a search space of small dimensionality, potentially biasing experimental results. We employ a conventional, stochastic backpropagation algorithm for pattern-by-pattern learning with a constant learning rate of η=0.5 without momentum for a maximum of 1000 epochs, using early stopping if the MSE has not improved by 1% in the last 50 epochs. The naive setting both benefits unwanted convergence and entrapment into local minima and premature stopping due to slow learning convergence. The resulting experimental design creates a large number of possible candidate models, many of them overfitted on either training or validation set, which creates problems in model selection (similar as for SVR).

To summarise, we combine a number of common mistakes in MLP modelling to create a Naive methodology. This methodology may serve as a lower bound of performance to all methodologies developed for the class of MLPs, and hence an objective benchmark for the NNGC competition.

D. Naive Support Vector Regression

In this study we seek to explore the simplest grid-based approach, using a brute-force, exhaustive enumeration of a representative parameter space. Furthermore, we limit the choice of kernel function to radial basis kernels (RBF), as it is most commonly used in ε-SVR using just one parameter γ to determine the kernel width a priori [1]. The number of centres, location of the centres, the weights of the RBF plus all thresholds are determined during training [26]. Similar to the MLP, we employ a simple grid search of costs C, epsilon ε and the width of a Gaussian kernel γ with exponentially growing sequences that cover a vast range of value combinations. Employing a grid search methodology requires the setting of valid and reliable lower and upper parameter bounds that define the search space of the grid. To follow an exhaustive approach, we set the lower bound to $C_l=2^{-10}$ and the upper bound to $C_l=2^{16}$, exceeding the parameter range of previous experiments (see section II.C) in order to provide the capability for a sufficient trade off for the different time series patterns. This implements an exponential grid with 36 steps of $2^{0.5}$ to evaluate the parameter values of C=[2^{-10},$2^{-9,5}$,…,2^{16}]. Similarly, for the parameter ε that controls the size of the ε-insensitive tube and the number of support vectors we extend these search spaces of previous studies and use a lower margin of 2^{-8} with an upper margin of 2^0, employing exponential grid steps of $2^{0.25}$, evaluating 32 parameter values of ε=[2^{-8},$2^{-7,75}$,…,2^0] for different noise. For the kernel width γ we select an exponential grid with steps of $2^{0.5}$, evaluating 30 parameter values of γ=[2^{-12},$2^{-11,5}$,…,2^0] to provide feasible kernel parameters for the scaled time series data, again exceeding limits of prior studies. Due to the magnitude of this time intensive approach of parameter selection, we reduce the training time by applying a shrinking technique to speed up the decomposition used to solve the SVR optimization problem, iteratively removing bounded components so that reduced problems are solved (see [33] for details). All experiments are calculated using the LIBSVM libraries [34] and Intelligent Forecaster (www.bis-lab.com).

As a result, we evaluate thousands of ε-SVR candidate models on all time series of the competition. The evaluation of a wide range of parameter combinations for each time series by grid search, without a robust methodology nor coordination of interacting parameters, facilitates various problems in model building, in particular overfitting through excessive parameter combinations which increase the probability of fitting only on the validation performance, and increased time and decreased efficiency in modelling. Furthermore, this amplifies the problem of model selection

of a single best candidate from thousands on small cross-validation datasets, as discussed in the next section. As a consequence, the resulting SVR candidates may be considered a Naive model building approach.

E. Model Selection

Depending on the flexibility of model parameters, both MLP and SVR are capable of approximating the underlying data generating process of a time series to different degrees of accuracy, permitting overfitting to the training data though a combination of sub-optimal parameters and thereby limiting its ability to generalize on unseen data [4] Hence the selection of a robust model candidate for each time series requires particular attention. To select the 'best' MLP or SVR candidate model from the different parameter setups, each time series is split into two subsets of 65% training data and 35% validation data for single fold cross validation [35]. Considering the length of the series the validation set is selected to roughly match the undisclosed test set in length, serving as a first estimate of a quasi-out-of-sample accuracy. Each candidate model is parameterised on the training dataset and is selected exclusively on its validation dataset.

As only a short validation dataset is used for selecting the best candidate model for that time series, overfitting on the validation set frequently occurs if the validation subset does not fully represent the true data generating process, which cannot be expected from small data sub-samples. Multiple approaches are feasible to avoid overfitting to the validation data in model selection and to derive an unbiased estimator on unseen data, including methods for data sub-sampling such as *k*-fold cross validation using different numbers of data folds or leave-one-out cross validation [36].

To adhere to a naïve approach of model building, short of avoiding the grave mistake of selection of the best candidate model on the in-sample training data itself, we compute only single cross-validation errors and select the best model on the prefixed validation set. Consequently all MLP and SVR candidates are parameterised exclusively on the training set, while the forecasting capability of the models is evaluated on the validation set and the candidate model with the lowest 1-step ahead validation error is selected [37].

Empirical simulation experiments have proven that error measures play an important role in calibrating and refining, model selection and ex post evaluation of forecasting models in order to determine the competitive accuracy and rank candidate models [37-38]. Although they should be selected with care, we apply the suboptimal squared error loss function of the root mean squared error (RMSE), weighting each error deviation by the quadratic distance using:

$$RMSE = \frac{1}{n}\sqrt{\left(\sum_{t=1}^{n}\left(e_t\right)^2\right)} \tag{4}$$

Using quadratic error or l_2-loss emphasises the influence of large forecast errors over small ones, e.g. from outliers and missing values. Evidence in forecasting literature confirms that squared error loss should normally be avoided in the evaluation of model performance, although practitioners and academicians in CI regularly employ MSE and RMSE to draw conclusions about forecasting methods [38], biasing the objective and evaluation of the algorithms. Although squared error metrics are frequently used due to their established history in conventional least-squares-estimators and their mathematical simplicity, as the selection criteria they also diverge from the final forecasting error metric in the NNGC competition of the symmetric mean absolute percent error (SMAPE) [38-39] which introduces a further mismatch:

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|y_t - \hat{y}_t\right|}{\left(\left|y_t\right| + \left|\hat{y}_t\right|\right)/2}\cdot(100) \tag{5}$$

The model with the lowest RMSE on forecasting multiple t+1 step-ahead forecasts on the validation set is selected and applied to predict the next h data points as multiple-step-ahead forecasts t+1, t+2, …, t+18 on the NN3 competition data sets. It is apparent, that this gives rise to another mismatch, as a method may show adequate accuracy on forecasting one step into the future, yet another set of parameters may perform better in forecasting multiple steps ahead. As this is commonly not aligned in previous studies, we comply with this malpractice in the naïve methodology, introducing further potential for misspecification errors. In addition to model selection, errors from multiple-step-ahead trace forecasts may also be used for error backpropagation and early stopping in training, in accordance with the forecasting horizon h rather than mere 1-step ahead forecasts, rolled forward by each time origin to achieve rolling origin evaluation[40]. This could aid in further aligning the forecasting objective and model specification.

V. Experimental Results

No true observations for the test data of the NNGC competition are available at this time, so no evaluation of out-of-sample accuracy may be conducted. More thorough evaluations of the naïve methodology would be feasible by splitting the available data into training, validation and test set, but these are not conducted due to the obvious sub-optimality of the naïve approach. This limits the following investigation to an analysis of the relative performance of the Naive benchmarks of SVR and MLP in comparison to more sophisticated methodologies submitted by contestants, but only once the final results of the NNGC are published.

VI. Conclusions

We compute a naïve heuristic, making use of most frequent mistakes in MLP and SVR modeling for time series prediction in order to establish a lower bound of accuracy for the respective model classes in the NNGC competition. The naïve heuristics evaluate an extensive grid search of MLP and SVR parameter combinations for each time series,

calculating thousands of candidate models with the ensuing problems in model selection. In aiming for a lower bound, we neglect the necessity to identify a significant input vector per time series, evaluate different scaling schemes, evaluate different algorithm specific parameters such as activation or kernel functions, control for overfitting in model selection form the validation data using *k*-fold cross-validation, conducting model selection and evaluation on a representative error metric for the ex post evaluation of the performance or the true cost of the decision, and computing and evaluating one step ahead predictors instead of multiple-step-ahead predictors as required in the final test evaluation. The naïve heuristic identifies a set of parameters for each of the 11 time series of the 6 datasets, which are subsequently used to forecast the next *h* future steps for unseen data.

While we hope to demonstrate the general ability of MLP and SVR to forecast linear and nonlinear time series with seasonal and non-seasonal patterns, showing their comparatively robust performance despite purposeful model misspecification, the discussion of the naïve heuristic methodology also aims at drawing attention to the most common mistakes in MLP and SVR model building. Further research, and systematic evaluations of forecasting accuracy on multiple empirical time series are required to establish a valid, reliable and robust methodology for automatic MLP and SVR model building. For this, we hope that forecasting competitions such as the NNGC on empirical data serve as a valid and objective initial test bed. Until then, the naïve grid search heuristic may serve not only as a negative benchmark and lower bound to forecasting accuracy, but also as a warning on the most obvious pitfalls to avoid in MLP and SVR model building.

REFERENCES

[1] C.-H. Wu;, *et al.*, "Travel Time Prediction with SVR," in *IEEE Intelligent Transportation Systems Conference*, 2003, pp. 1438- 1442.

[2] H. Yang;, *et al.*, "Outliers Treatment in SVR for Financial Time SeriesPrediction" *Computer Science,* vol. 3316, pp. 1260-1265, 2004.

[3] G. Zhang, *et al.*, "Forecasting with artificial neural networks: The state of the art," *I.J. of Forecasting,* vol. 14, pp. 35-62, 1998.

[4] A. J. Smola; and B. Schölkopf, "A Tutorial on Support Vector Regression," *Statistics and Computing,* vol. 14, pp. 199-222, 2004.

[5] K.-P. Liao; and R. Fildes, "The Accuracy of a Procedural Approach to Specifying Feedforward Neural Networks for Forecasting," *Computers & Operations Research,* vol. 32, pp. 2121-2169, 2005.

[6] G. Lachtermacher and J. D. Fuller, "Backpropagation in Time-Series Forecasting," *Journal of Forecasting,* vol. 14, pp. 381-393, Jul 1995.

[7] G. E. P. Box, *et al.*, *Time series analysis : forecasting and control*, 3. ed ed. Englewood Cliffs, NJ [u.a.]: Prentice Hall, 1994.

[8] S. S. Haykin, *Neural networks : a comprehensive foundation*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.

[9] K. P. Liao and R. Fildes, "The accuracy of a procedural approach to specifying feedforward neural networks for forecasting," *Computers & Operations Research,* vol. 32, pp. 2151-2169, Aug 2005.

[10] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing,* vol. 10, pp. 215-236, 1996.

[11] R. Neuneier and H.-G. Zimmermann, "How to Train Neural Networks," in *Neural networks : tricks of the trade*, G. Orr and K.-R. Müller, Eds., ed Berlin ; New York: Springer, 1998, pp. 373-423.

[12] Z. Y. Tang and P. A. Fishwick, "Feed-forward Neural Nets as Models for Time Series Forecasting," *ORSA JoC,* vol. 5, pp. 374-386, 1993.

[13] M. Adya and F. Collopy, "How effective are neural networks at forecasting and prediction? A review and evaluation," *Journal of Forecasting,* vol. 17, pp. 481-495, Sep-Nov 1998.

[14] S. D. Balkin and J. K. Ord, "Automatic neural network modeling for univariate time series," *IJF,* vol. 16, pp. 509-515, 2000.

[15] M. C. Medeiros, *et al.*, "Building neural network models for time series," *Journal of Forecasting,* vol. 25, pp. 49-75, Jan 2006.

[16] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence,* vol. 97, pp. 273-324, 1997.

[17] V. N. Vapnik, "An Overview of Statistical Learning Theory," *IEEE Transactions on Neural Networks* vol. 10, pp. 988-1000, 1999.

[18] N. Cristianini; and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based Learning Methods*. Cambridge (United Kingdom): Cambridge University Press, 2000.

[19] A. Smola, "Regression Estimation with Support Vector Learning Machines," Diplomarbeit, Technische Universität München, 1996.

[20] K.-R. Müller; *et al.*, "Predicting Time Series with SVM," in *Advances in Kernel Methods — Support Vector Learning*, B. Schölkopf;, *et al.*, Eds., ed Cambridge: MIT Press, 1999, pp. 243-254.

[21] V. Cherkassky; and Y. Ma, "Practical Selection of SVM Parameters and Noise Estimation for SVM Regression," *Neural Networks,* vol. 17, pp. 113-126, 2004.

[22] S. F. Crone; *et al.*, "Parameter Sensitivity of Support Vector Regression and Neural Networks for Forecasting," in *International Conference on Data Mining*, Las Vegas (U.S.A.), 2006.

[23] M.-W. Chang; and C.-J. Lin, "Leave-One-Out Bounds for SVR Model Selection," *Neural Computation,* vol. 17, pp. 1188-1222, 2005.

[24] J. T. Kwok; and I. W. Tsang, "Linear Dependency Between epsilon and the Input Noise in epsilon -SVR," *IEEE Transactions on Neural Networks,* vol. ICANN 2001, pp. 405-410, 2003.

[25] A. Smola;, *et al.*, "Asymptotically optimal choice of epsilon-loss for support vector machines," in *Proceeding of the International Conference on Artificial Neural Network*, 1998.

[26] C. J. C. Burges, "A Tutorial on SVM for Pattern Recognition," in *Data Mining and Knowledge Discovery*. vol. 2, U. Fayyad, Ed., ed Boston (U.S.A.): Kluwer Academic Publishers, 1998, pp. 121-167.

[27] J. B. Gao;, *et al.*, "A probabilistic framework for SVM regression and error bar estimation," *Machine Learning,* vol. 46, pp. 71-89, 2002.

[28] W. Chu;, *et al.*, "Bayesian SVR Using a Unified Loss function," *IEEE Transactions on Neural Networks* vol. 15, pp. 29-44, November 2002.

[29] C.-J. Lin; and R. C. Weng, "Simple probabilistic predictions for support vector regression," National Taiwan University, Taipei2004.

[30] C.-W. Hsu;, *et al.*, "A Practical Guide to Support Vector Classification," National Tawain University Taipei (Taiwan)2003.

[31] C.-W. Hsu; and C.-J. Lin, "A Comparison of Methods for Multiclass SVM" *IEEE TNN*vol. 13, pp. 415-425, 2002.

[32] V. R. Prybutok, *et al.*, "Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone," *EJOR,* vol. 122, pp. 31-40, Apr 2000.

[33] R.-E. Fan;, *et al.*, "Working Set Selection Using Second Order Information for Training SVM," *JMLR,* vol. 6, pp. 1889-1918, 2005.

[34] C.-C. Chang; and C.-J. Lin, "LIBSVM: a Library for SVM" National Science Council of Taiwan, Taipei (Taiwan)17. April 2005.

[35] P. Cunningham, "Overfitting and Diversity in Classification Ensembles based on Feature Selection," Trinity College Dublin, Dublin (Ireland), Technical Report: TCD-CS-2000-07, 2000.

[36] D. Opitz; and R. Maclin, "Popular Ensemble Methods: An Empirical Study," *JAIR,* vol. 11, pp. 169-198, 1999.

[37] S. Makridakis;, *et al.*, *Forecasting Methods and Applications*, 3 ed. New York: John Wiley & Sons, 1998.

[38] J. S. Armstrong; and F. Collopy, "Error Measures for Generalizing About Forecasting Methods," *IJF,* vol. 8, pp. 69-80, 1992.

[39] R. J. Hyndman; and A. B. Koehler, "Another Look at Measures of Forecast Accuracy" Monash University, Working Paper 13/05, 2005.

[40] L. Tashman, "Out-of-sample tests of forecasting accuracy: an analysis and review," *International Journal of Forecasting,* vol. 16, pp. 437-450, 2000.