

Forecast Combinations of Computational Intelligence and Linear Models for the NN5 Time Series Forecasting competition *

Robert R. Andrawis
Dept Computer Engineering
Cairo University, Giza, Egypt
robertrezk@yahoo.ca

Amir F. Atiya
Dept Computer Engineering
Cairo University, Giza, Egypt
amir@alumni.caltech.edu

Hisham El-Shishiny
IBM Center for Advanced Studies in Cairo
IBM Cairo Technology Development Center
Giza, Egypt
shishiny@eg.ibm.com

November 6, 2010

Abstract

In this work we introduce a forecasting model with which we participated in the NN5 forecasting competition (the forecasting of 111 time series representing daily cash withdrawal amounts at ATM machines). The main idea of this

*Accepted for publication in International Journal of Forecasting. Expected to appear in 2011.

model is to utilize the concept of forecast combination, which has been proven in the forecasting literature as an effective methodology. In the proposed system we attempted to follow a principled approach and make use of some of the guidelines and concepts that are known in the forecasting literature to lead to superior performance. For example, we considered some of the comparison studies and some of the previous time series competitions as guidelines to determine which individual forecasting models to test (for the possible inclusion in the forecast combination system). The final model ended up consisting of neural networks, Gaussian process regression, and linear models, combined by simple average. We also paid extra attention to the seasonality aspect, decomposing the seasonality into weekly seasonality (which is the strongest one), day of the month seasonality, and month of the year seasonality.

1 Introduction

In this study we present the model with which we participated in the NN5 time series competition [46]. Our model achieved for the complete data set the first rank out of 19 competitors, among the computational intelligence models, and the second rank out of 27 competitors overall (that is among computational intelligence models and statistical benchmarks). For the reduced data set our model obtained the third rank out of 29 competitors among the computational intelligence models, and the fourth rank out of 39 competitors overall (see Crone [15] and [46]). In the NN5 competition the task has been to forecast 111 daily time series, representing daily cash withdrawal amounts at ATM machines in various cities in the UK. The length of each time series is two years, and the forecast horizon is 56 days.

We attempted to follow a principled approach, whereby we use concepts and methods that are known from the literature to lead to superior performance. In situations where no dominant method is known we have experimented with various methods. For example, it has been an accepted fact that combining multiple forecasts tends to improve the forecasting performance, often in a dramatic way. Therefore, a major aspect of our model is the use of multiple forecasts and combining them. Since a forecast combination system inherits its forecasting performance from its constituent forecasting models, it was essential for us to experiment with many forecasting models, and select the best few to combine. There is essentially a very large pool of potential computational intelligence based models to test. So we limited the search to models that are known from previous comparison studies and from previous time series competitions to be good performers. Some of the computational intelligence models that ended up being selected in the final model are neural networks and Gaussian process regression. While the superiority of neural networks has been demonstrated in several studies, it was of interest to find among the top models the Gaussian process

regression, a hitherto not very commonly studied model in the forecasting literature. Gaussian process regression is basically a nonparametric regression model based on a probabilistic formulation of the concept of spatial smoothness. We also considered some linear models, as simple models are known to yield good forecasting results. This also provides an extra diversification away from nonlinear models. Some of the linear models that ended up being selected are multiple regression models and a simple moving average model. We also tested different time series preprocessing and postprocessing methods. For example, we considered several ways of carrying out the multistep ahead forecasting, such as the direct and the iterative methods. Also, we analyzed different time aggregations of the time series. All these variations have helped in adding diversity to the forecast combination model, as well as tuning in on the best models to be used.

A third aspect to which we paid extra attention is the seasonality aspect of the time series. As far as we know, the majority of the research on the seasonality topic (for example Zhang and Kline [57], and Zhang and Qi [56]) and the seasonality software packages (such as X12-ARIMA of the U. S. Bureau of the Census) handle monthly or quarterly time series. The problem considered here is somewhat atypical. We have daily time series with strong day of the week seasonality, moderate day of the month seasonality, and also moderate month-of-the-year seasonality. We have modified the well-known seasonal decomposition approach, by using the median instead of the average, and used the group seasonality concept for some of the seasonal components.

The paper is organized as follows. Next section we present a brief overview of the forecast combination problem. Section 3 describes the experimental design, clarifying the process of testing forecasting models for possible inclusion in the forecast combination system. Section 4 presents details about the time aggregation performed, while Section 5 presents the considered methods for performing multistep ahead forecasting. Section 6 is a detailed analysis of the seasonality aspects of the time series. Section 7 presents the final selected models that we ended up including in the forecast combination system. Section 8 gives a detailed analysis of the developed model and its forecasting results. Finally, Section 9 gives a summary and a conclusion for this work.

2 Forecast Combination

It has been documented in the literature that forecast combinations are very often superior to their constituent forecasts (Timmermann [54]). The alternative to forecast combination is to test several forecasting models, estimate their parameters, and select the model giving best performance in the in-sample period. However, practical experience has shown that the best model in the in-sample period might not be

best when forecasting future values. Typically, time series encounter time varying conditions, or possibly an out-right regime switch. This problem is also aggravated by parameter estimation errors and model misspecification. To hedge against all these unfavorable effects, a good strategy is to develop several good forecasting models, and combine their forecasts. In several studies (such as Clemen [14], Makridakis and Hibon [43], and Stock and Watson [51]) combined forecasts have generally been shown to outperform the forecast from the single best model. A theoretical justification of forecast combination can be obtained by viewing the problem from a Bayesian model averaging perspective (Hoeting et al [33]). Without knowledge of the precise data generating process of the time series, several models have to be tested, and their forecasts averaged according to the likelihood of the underlying models.

Forecast combination was pioneered in the sixties by Bates and Granger [9] and Ried [49]. Since then many developments occurred in this field, and several review articles have appeared (Clemen [14], DeMenezes et al [17], Diebold and Lopez [20], and Timmermann [54]). Forecast combination was also introduced in the neural network field (see Hashem et al [29], [30]). A parallel though somewhat different track in the neural network field is the so-called ensemble approach (Dietterich [21]). In this approach many networks are trained under varying conditions, and their outputs are combined (examples of ensemble networks are the bagging method and the boosting method, see Dietterich [21]). Another related methodology is the so-called thick modeling approach (Granger and Jeon [28]). In this approach, rather than tuning precise values for the neural network parameters, several plausible sets of parameters are used and the outputs of their respective networks are combined.

Over the years through practical experience or through large scale comparison studies researchers have accumulated quite a bit of valuable knowledge and understanding of the forecast combination approach. We summarize briefly some of the findings and practical aspects. One of the favorable features to have in a forecast combination system is the diversity of the underlying forecasting models, as a hedge from being too focused on a narrow specification. The diversity could stem from using different forecasting models, different model specifications, different time series preprocessing, or different exogenous input variables. This allows us to “cover a lot of ground” in explaining the considered time series.

Another aspect is that simple combination schemes tend to perform best. Adding an extra parameter (even an intercept, for example) generally tends to harm the performance. Also, some approaches, such as shrinkage, that temper away some of the complexity, have been found to give favorable performance (Timmermann [54]). Makridakis et al [41] and Clemen [14] have found that the simple average (i.e. equal weight combination) performs the best or near the best compared to other combining methods. De Menezes et al [17] have reviewed several studies and have come to the conclusion that the simple average outperforms when the individual forecasts are

comparable in performance, otherwise other methods will be more superior. Trimming out bad forecasting models also tends to improve performance (Timmermann [54]). Also, Aiolfi and Timmermann [3] have found evidence of persistence in out of sample performance for linear and nonlinear forecasting models. Therefore, for a forecast combination system to be superior, the underlying forecasting models have to be good performers (using the performance in the evaluation or validation set as a guide). In other words, a forecast combination system will not make up for poor performance in some of the underlying models.

In our developed model, we followed these guidelines above, while performing some experimentation to verify the suitability of our choices. Concerning diversity, in our model we imparted diversity by combining nonlinear models (computational intelligence based) and linear models (naturally linear and nonlinear models are inherently different). Also, we achieved diversity by considering a varying group of computational intelligence models, different combinations of input variables, and different approaches for tackling the multistep ahead problem (such as the direct approach, the iterative approach, and the parallel approach, see next section for a description).

As mentioned, it is essential that the constituent forecasting models be as good as possible. In our system we attempted to achieve this property by experimenting with a large number of potential candidates. From all tested models only the top nine models are used in the ultimate combination system. Armstrong [6] advocates generally using five or six models. In our case the number nine is chosen because nine models achieved top and comparable performance. Taking more than that would have led to including some of the lesser performing models that could possibly be a drag on the overall performance. Concerning the combination weights, we opted for the simple average. Since the nine constituent forecasts are comparable in performance, the simple average is the most reasonable way to go.

3 Model Pre-Testing

Before developing the forecasting models an essential first step is to deseasonalize the time series (described in more details in Section 6). The considered problem is a hard 56-day ahead forecasting problem. To simplify the problem, we performed a time aggregation step to convert the time series into a weekly series. The details of all performed time aggregations are given in the next section.

The next step, discussed here, is to pre-screen the models. This means that we compare between a large number of computational intelligence and linear models, and decide upon the best few. As we mentioned in the previous section, our aim is to determine a few very good forecasting models, and combine their forecasts.

We experimented with a total of about 140 different models. Each one corresponds

to either a different computational intelligence or a linear model, or a different preprocessing or postprocessing method. The following are the classes of models we considered.

1. Standard multilayer neural network.
2. Gaussian process regression.
3. Echo state network.
4. Echo state network ensemble.
5. ARMA (with a variety of order specifications).
6. AR (with order specification using the BIC criterion).
7. Multiple regression.
8. A version of Holt’s exponential smoothing by Andrawis and Atiya [5] whereby the parameters are obtained in a maximum likelihood framework.
9. Simple moving average.

For each of these models several variations of model parameters, preprocessing, and postprocessing are considered, leading to a total of 140 models. For example, we considered different combinations, aggregations, or numbers of input variables (see next section for a description). Also, we considered different ways to perform the multi-step ahead forecasting (see Section 5). Other preprocessing that we tested includes applying a log transformation to the time series, and detrending the time series. Both of these, however, were not very beneficial, and ultimately ended up not being selected in the final systems.

The selection of the final models to use in the forecast combination system is purely based on forecasting accuracy (on the test set). We used 97 weeks as training data and 8 weeks as a test set. The length of the test period is set so as to replicate the conditions of the period to be forecasted (as set out in the competition). Once the models are finalized, we combine the training set with the test set to retrain the models and then generate the final forecast (to be submitted to the competition).

The reason for considering the pool of forecasting models listed above is as follows. A previous large scale study by Ahmed et al [2] compared between eight computational intelligence models for time series forecasting. The top two models turned out to be the standard neural network model and the Gaussian process regression model. We had also applied these two models in several other forecasting applications and verified their strong performance. We also checked the results of the NN3 time series

forecasting competition (the precursor competition to this NN5 competition), and found that the echo state network ESN (a type of recurrent neural network) was the top model in the computational intelligence category. So we considered several variations of ESN's, including an ensemble of ESN's (since ensemble networks are generally known from many studies to be quite successful). As can also be seen in the list, we included a variety of the major linear models, to diversify away from computational intelligence models which are all nonlinear. Also, simple models tend to fare well in forecasting applications, especially in very noisy applications that have small data sets.

4 Time Aggregation and Input Variables

To simplify the forecasting problem, we performed a time aggregation step to convert the time series from daily to weekly. We considered the weekly version of the time series (where each weekly data point is constructed by summing the underlying daily data points). So, in this case, the forecast becomes simply an 8-step-ahead forecast (that is forecasting 8 weeks), which makes the problem more manageable. Once the forecast has been performed, we convert the weekly forecast to a daily one by a simple linear interpolation scheme. This scheme has the following conditions: a) the forecast of the week equals the sum of the constituent daily forecasts; b) the daily forecasts for a specific week follow a straight (possibly sloping) line; c) from one week to the next the line's slope could change, generally leading to a piecewise linear function; d) furthermore this piecewise linear function is a continuous function, that is the ending point for one week is the starting point for drawing the line for the next week. The computation for performing this translation from weekly forecasts to daily forecasts is fairly simply performed.

The preprocessing and postprocessing varieties that we tested included giving different combinations of input variables. Most of the input variables given were lagged values and window averages for the weekly time series. For example, the input variables to a model could be the k previous weekly values. We tested various values of k . When going beyond four weeks of lagged values, we used coarser time aggregation, typically monthly. For example we could have the four weeks lagged values, and the averages of each of the two months previous to these four weeks. (A useful study of time aggregations for neural network forecasting can be found in Atiya et al [7].)

When we mentioned about the time aggregation step (converting the daily time series into a weekly time series), we made a point that the problem becomes more manageable (forecasting 8 points instead of 56 points). However, one might wonder, perhaps performing some further aggregation might lead to improvements, or at least could lead to more diverse models. We focused on this issue, but only for the horizon

to be forecasted. That is, we considered aggregating that only, but not the input variables entering into the models. By inspecting the time series, we found that there are not many significant medium term trends. The values for the time series are not expected to vary much beyond that exhibited by the seasonalities (and once in a while some level shifts). We therefore have a model that is trained to forecast the whole eight week horizon as a fixed constant value. The issue is here to estimate the constant level that is expected to best fit the future time series values (abbreviate this approach as LEV). We considered also an approach where we have the time aggregation varying with the horizon. For example the first few days in the forecast horizon are very close to the current day and therefore previous variations at the scale of days will have an impact. On the other hand, beyond these adjacent days a coarser aggregation could be more appropriate. We therefore tested the following time aggregation approach. The first seven days in the forecasting horizon will be forecasted separately (no aggregation is done, they are kept in their daily form). The lagged time series values entering as input variables will also be the past daily lags. Beyond these seven days, we use weekly aggregation or more.

5 Methods for Multi-Step Ahead Forecasting

We considered different ways of carrying out the multi-step ahead forecasting. There are three well-known ways: the iterative approach, the direct approach, and the parallel approach. A thorough analysis and comparison of these approaches can be found in Kline [39].

In the iterative approach (abbreviate it as ITER) the model is trained on a one-step ahead basis. After training we use the model to forecast one step ahead (i.e. one week ahead). We then use the forecasted value as input to the model when we forecast the subsequent point, and continue in this manner. In the direct approach (abbreviate it as DIR) we use a different network for every point in the future to be forecasted. For example, in our case we have an eight week forecast horizon, so we have eight networks, each one forecasting a specific week in the future. Of course, every network is trained separately. When forecasting, we apply these eight networks to obtain the eight weeks ahead. In the parallel approach, we use only one network with a number of outputs equal to the length of the horizon to be forecasted. We train the network in a way so that output number k produces the k^{th} step ahead forecast. We focused mainly on the iterative approach and the direct approach. The parallel approach could apply only to the neural network and the echo state network (because they allow multi-output), and the initial runs did not yield good performance for this approach.

6 Seasonality Analysis

In the business forecasting literature there has been two competing views in handling seasonality. In the first one the seasonal component is estimated and factored out of the time series. Then, a forecasting model is applied on this *deseasonalized* time series. In the other view a forecasting model is designed to outright take into account the seasonal aspect of the time series. Examples of this latter approach are the seasonal ARIMA model (or SARIMA), and the Holt-Winters exponential smoothing model [24]. There has been some debate as to which approach is more effective. The advantage of the seasonal adjustment approach is that it decomposes the problem and therefore allows the forecasting model to pay more attention to the other aspects of the time series such as the trend. The opponents, however, argue that deseasonalization sometimes unfavorably alters the characteristics of the time series. For example, a transformed data point includes future values of the time series due to the typically used centered moving average, leading to some kind of “forward looking” when testing or tuning a forecasting model on the in-sample period. Nevertheless, deseasonalization typically leads to favorable results. For example Wheelwright and Makridakis [55] found that a deseasonalization step led to improved accuracy for traditional statistical forecasting models. In the computational intelligence literature, deseasonalization was mostly a beneficial strategy. Nelson et al [44] considered neural network forecasting models applied on 68 monthly time series from the M-competition. They showed that a deseasonalization step led to a significant improvement in forecasting performance. Zhang and Qi [56] and Zhang and Kline [57] arrived at the same conclusion, and advocate that deseasonalization (as well as detrending) is very beneficial when using neural network forecasting models. Therefore, in our study, we opted for the deseasonalization approach.

The prevalent way to extract seasonal averages is the well-known additive or multiplicative seasonal decomposition approach (Makridakis et al [42]). In this method a centered moving average is applied, then the time series is divided by this moving average. The seasonal average will then be the season-by-season average of the resulting series. A group of deseasonalization systems called collectively X11 systems include the well-known X11, X11-ARIMA, X12-ARIMA, and TRAMO/SEATS systems. They are comprehensive systems that utilize a group of moving averages, applied in an iterative multi-step way. They also include other aspects such as outliers handling and some diagnostics (see Ghysels [25] and Gomez and Maravall [27]). There have also been other approaches to extract the seasonal component, for example by regressing the time series on a number of sine and cosine functions (Hylleberg [37]), to extract the periodic component.

The approach that we used is essentially similar to the multiplicative seasonal decomposition, except that we use the median instead of the mean to compute the

seasonal average. The time series in this competition exhibit strong seasonal components at a variety of time scales. The strongest seasonal component is the day of the week seasonality. The weekday will naturally have an impact on withdrawal volume, due to weekend effects and the generally varying weekday patterns of people's behavior. Another seasonal effect is the day of the month seasonality. Naturally wage payments and bill payments coincide with certain days of the month. The final seasonal effect is the month of the year. Holiday and back-to-school months generally exhibit higher withdrawal volume.

To test the existence of these three types of seasonality in the different time series, we employed the following tests. The first one tests for day of the week seasonality. In this test, we normalize each time series value by dividing by the average of the week containing it. Then we group the data by day of the week and perform a one-way ANOVA test for comparing the means of the seven groups of data (representing the the seven days of the week). This test rejected the null hypothesis (of equal group means) and therefore indicated that all time series possess a day of the week seasonality (with an essentially zero p-value). Figure 1 shows a box-plot for the average day of the week seasonal pattern for one of the time series. The second test is an analogous test, but applied to the day of the month seasonality case. The test did not reject the null hypothesis (at the 5% level) for any of the 111 time series. Another look at the data revealed that this is due to the scarcity of the data points (typically around 24 points per time series). However we noticed that there is a pattern of high withdrawals in the first week of the month and at mid-month, and a clear trough in the last week of the month. This same pattern runs across different time series, suggesting the possible existence of a group seasonality. To investigate that, we performed the following test. We considered each of the normalized time series (normalized by dividing by the average of the month), and computed the mean for each day of the month (the mean across all months of the considered time series). So we ended up with 31 numbers representing the mean values for every time series. Considering these numbers for all time series as a whole and performing an ANOVA test, we tested the null hypothesis: are the data for all time series grouped by day of the month of equal means. The test rejected the null hypothesis (with $p=0$) indicating the existence of day of the month group seasonality. Figure 2 shows the box-plot for the day of the month group seasonal variation. We also tested for month of the year group seasonality, by creating a normalized monthly time series (each time series point would be the average of the month normalized by the average of the year). Again, the ANOVA test rejected the null hypothesis of equal means for each of the 12 months (over all time series), indicating the existence of group month of the year seasonal behavior. Figure 3 shows the box-plot for the month of the year group seasonal variation.

Now that we established the existence of individual day of the week seasonality,

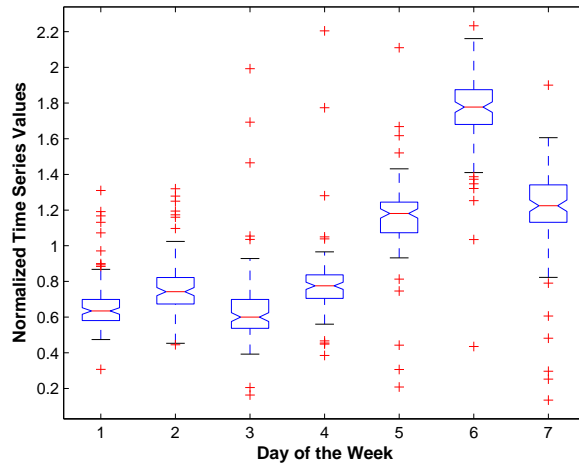


Figure 1: A box plot of the day of the week seasonal average for one of the time series.

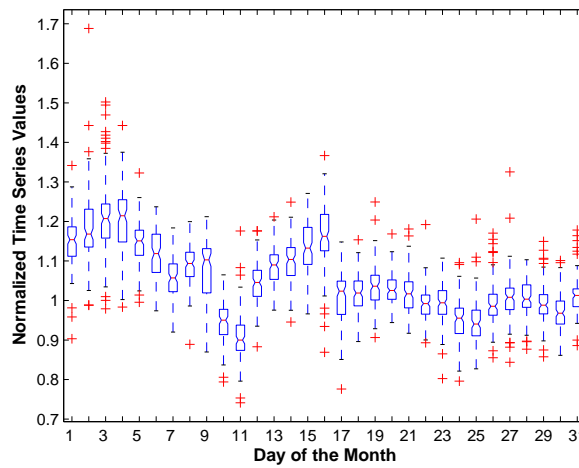


Figure 2: A box plot of the day of the month group seasonal pattern for all time series.

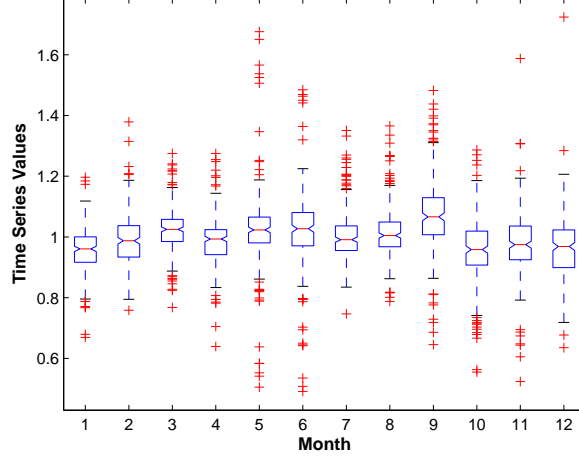


Figure 3: A box plot of the month of the year group seasonal pattern for all time series.

and group day of the month and month of the year seasonalities, we proceed with the deseasonalization step. For the day of the week seasonality, the method is described as follows:

1. For every time series point compute the average value for the week containing it. Denote this by $AvgWk$.
2. Compute the relative or normalized time series value:

$$z_t(i) = \frac{y_t(i)}{AvgWk} \quad (1)$$

where $y_t(i)$ is the time series value at time t that is a day of the week i , and $z_t(i)$ is the normalized time series value assuming it is of day of the week i . This normalization step is designed to take away the effects of any trend or level shift that would affect the absolute values for the different days of the week.

3. Compute the seasonal average $s_W(i)$, as follows:

$$s_W(i) = \text{Median}_t(z_t(i)) \quad (2)$$

The median is computed over all points in the time series that are of day of the week i .

4. The deseasonalization is obtained as:

$$u_t(i) = \frac{y_t(i)}{s_W(i)} \quad (3)$$

where $u_t(i)$ represents the deseasonalized series.

As mentioned, a distinctive feature of this deseasonalization algorithm is the use of the median instead of average. Weekly seasonality involves in most cases sharp pulses. If using the average, the peak of the seasonal average becomes blunter and shorter, leading to detrimental results. The median, on the other hand, leads to preserving the typical shape of the peaks observed during the week. Note that Crum [16] also used the median in the seasonal computation, but only for computing the median over the averaging window, and not for computing the seasonal average.

Concerning the day of the month seasonality, we employed a deseasonalization that is similar to the day of the week deseasonalization (also using the median concept). We will not get into the details, but at the end we obtained a seasonal average $s_M(j)$, $j = 1, \dots, 31$, where j is the day of the month.

The group seasonal average is also obtained by taking a day by day median of the $s_M(j)$'s pertaining to the different time series (to get say $s_{MG}(j)$). Even though the statistical test established group seasonality but could not establish individual seasonality, we considered in the simulations both these two variants (the ‘‘individual’’ and the ‘‘group’’ seasonalities). The reason is that according to Nelson et al [44] the penalty of performing deseasonalization when there is no seasonality is minimal, but the penalty of not performing deseasonalization when there is seasonal behavior is quite large. Another reason is that this allows us to add beneficial diversity into the models. The group seasonal indexes will be less noisy than an individualized seasonal average due to using more data to estimate it. Its drawback is that it is less attentive to the individual variations from one time series to another. Chen and Boylan [13] performed a very interesting study of individual versus group seasonal analysis. They particularly highlight the powerful performance of group seasonal models.

For the month of the year seasonal effect we also computed the seasonal average using an approach similar to the previous two seasonal effects. Here we used only the group version, as there was little data to compute any individual seasonal average. The seasonal average is obtained by taking the month by month median over all normalized time series. Let $s_{YG}(k)$, $k = 1, \dots, 12$ be the seasonal average, where k is the month number.

Deseasonalizing with respect to the day of the month and the month of the year seasonal components can then be achieved by dividing $u_t(i)$ by $s_M(j)$ (or $s_{MG}(j)$) and by $s_{YG}(k)$. Note that the forecast combination step is performed at the very end, that is after returning back the seasonal components.

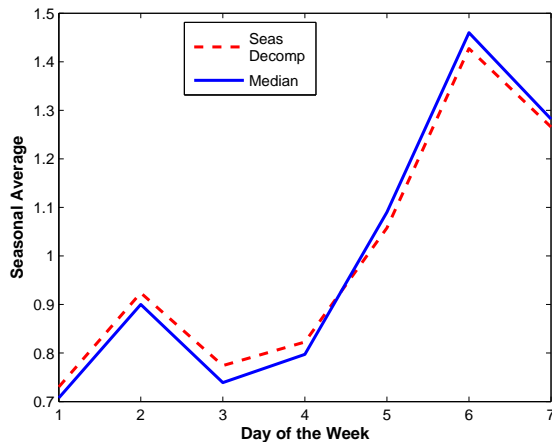


Figure 4: The day of the week seasonal averages for the median-based method and for the multiplicative seasonal decomposition for one of the time series.

Post-deseasonalization tests did not indicate statistically significant difference in the residual seasonality between the standard seasonal decomposition method and the method based on the median. However, the method based on the median led to significantly better ultimate forecasting performance, as indicated in the test of Section 8 (Table 2).

As an illustration, Figure 4 shows the weekly seasonal averages for one of the time series for the proposed median-based method as well as the multiplicative seasonal decomposition benchmark. As can be seen the median-based average has a higher and more defined peak. This characteristic (of a higher peak) was observed in 76% of the time series.

7 The Selected Models

7.1 The Nine Final Models

The nine models that gave best test forecasting accuracy turned out to be:

1. GPR-ITER: Gaussian process regression using the iterative multi-step ahead approach. The input variables to this model are the four weeks lagged values, the average of the month previous to these four weeks, and the average of the three weeks of the last year that correspond to the current week (this input is useful to account for any yearly seasonal effects).

2. GPR-DIR: Gaussian process regression using the direct multi-step ahead approach. The input variables to this model are similar to those of GPR-ITER.
3. GPR-LEV: Gaussian process regression using the fixed level multi-step ahead approach (i.e. the forecast of the whole 8 weeks is a fixed constant line, as described in Section 4). The inputs to this model are the four weeks lagged values, the averages of each of the two months previous to these four weeks.
4. NN-ITER: Neural network using the iterative multi-step ahead approach. The input variables to this model are similar to those of GPR-ITER.
5. NN-LEV: Neural network using the fixed level multi-step ahead approach. The inputs to this model are the four weeks lagged values, the averages of each of the two months previous to these four weeks.
6. MULT-REGR1: Multiple regression model, with varying time aggregation for the forecast horizon, as described in Section 4. Specifically, the first seven days are each forecasted separately, with the regressors being: the past seven days of the time series, and the average of the past 120 points. Beyond these seven days, we forecast the whole seven remaining weeks as a constant level. The regressors in this case are the four weeks lagged values, and the average of each of the three months previous to these four weeks (for a total of seven regressors). We used individual day of month seasonal average, rather than group.
7. MULT-REGR2: Another multiple regression model, very similar to MULT-REGR1 except that we used group day of month seasonal average, rather than individual.
8. MULT-REGR3: Another multiple regression model, very similar to MULT-REGR2, except that we used the average of each of the four months instead of three months as regressors.
9. MOV-AVG: A very simple model based on a simple moving average. The model forecasts the whole 8 weeks as a single level computed as the average of the previous 200 points.

All models are selected based on the accuracy on the test set. We used the *symmetric mean absolute percentage error* (SMAPE) as our error measure, as this is the main measure considered in the competition. It is defined as:

$$SMAPE = \frac{1}{M} \sum_{m=1}^M \frac{|\hat{y}_m - y_m|}{(|\hat{y}_m| + |y_m|)/2} \quad (4)$$

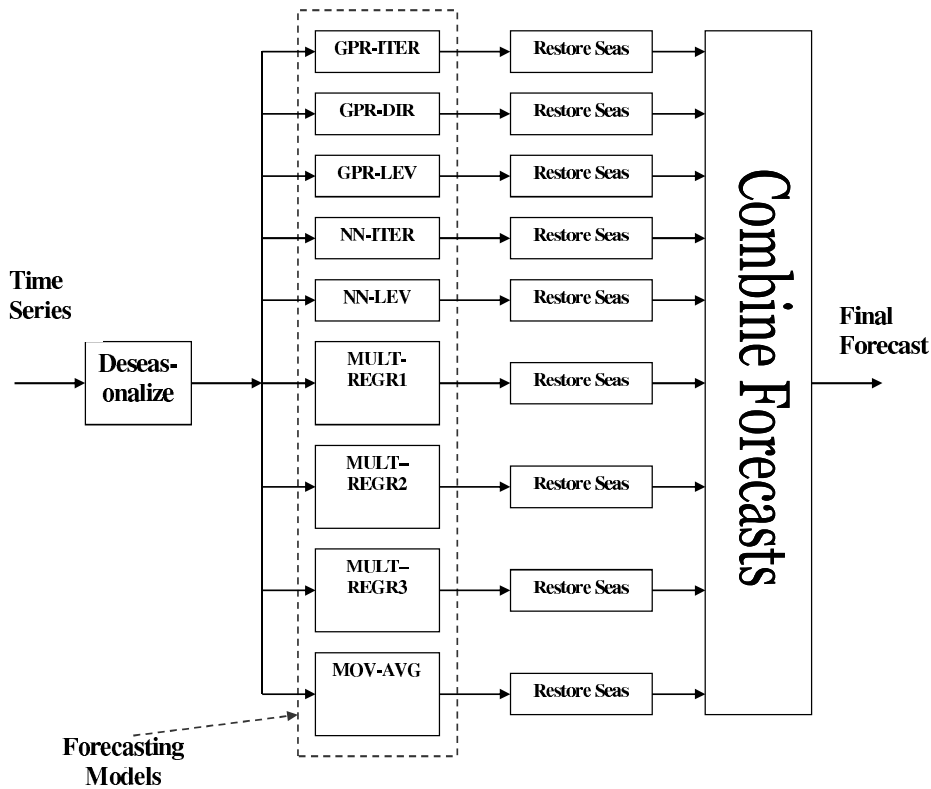


Figure 5: A block diagram for the developed forecasting system.

where y_m is the actual time series value and \hat{y}_m is the forecast, M is the size of the test period. All the tested 140 models (or model variations) yielded a widely varying range of test errors, with the SMAPE varying from around 19% to 40% (however the majority were in the range from 19% to 30%). The selected nine models (displayed above) are the top nine of the 140 candidate models. They gave very comparable errors, in the range 19% to 21%. Figure 5 shows a block diagram for the whole forecasting system, including the top models that ended up being part of the system. As mentioned, the forecasts of the nine selected models were combined by simple average, and the resulting forecasts were submitted to the competition.

Because the topic of the special issue is on computational intelligence models, we will describe in some brief detail below the neural network and the Gaussian process

regression models.

7.2 The Neural Network Model (NN)

The neural network (Hastie et al [31]). (often called multilayer perceptron) is a semi-parametric model, inspired by studies of the brain and the nervous system. It is an extremely flexible model, and in fact it has been proven to be a universal approximator. This means that given any continuous function one can always design a neural network that achieves an approximation as close as possible. A neural network consists of computational elements, called neurons, that perform a weighted sum operation, followed by applying a nonlinear squashing-type function. Specifically, the neural network output is given by:

$$\hat{y} = v_0 + \sum_{j=1}^{NH} v_j g(w_j^T x') \quad (5)$$

where x' is the input vector x , augmented with 1, i.e. $x' = (1, x^T)^T$, w_j is the weight vector for j^{th} hidden node, v_0, v_1, \dots, v_{NH} are the weights for the output node, and \hat{y} is the network output. The function g represents the hidden node output, and it is given in terms of the squashing function, for example the logistic function: $g(u) = 1/(1 + \exp(-u))$.

Neural networks have seen wide applicability in many diverse applications. In particular, they have been applied extensively to the time series forecasting problem, and their performance is quite encouraging (see for example Alon et al [4], Callen et al [10], Hill et al [18], Sharda and Patil [50], Terasvirta et al [53], Zhang and Qi [56], and the review Adya and Collopy [1]). Neural networks are trained using a gradient descent-type algorithm, called the backpropagation algorithm. More advanced optimization algorithms have also been widely used, such as second order optimization approaches. We used an algorithm from this group called Levenberg Marquardt, as this is generally known to be one of the more efficient algorithms for training neural networks (we use the Matlab function `trainlm`).

The number of hidden nodes NH , or network size, is a critical parameter for the NN, as it determines the network complexity. If the network size is not adequately controlled, the network can overfit the data. We have used five-fold cross validation to estimate this parameter for every time series. Kohavi [40] has shown that K-fold validation is one of the superior methods for model selection. We consider the candidate values $NH = [0, 1, 3, 5, 7]$. Note that we have the possibility of “having zero hidden nodes” ($NH = 0$), meaning simply a linear model. Balkin and Ord [8] have shown that the possibility of switching to a linear model for some time series improved

performance. We did not use multiple initializations, so for every different preprocessing/postprocessing combination we have exactly one neural network model. Instead of multiple initializations we used the initialization method by Nguyen-Widrow [45] (the Matlab function `initnw`). Concerning the other less key parameters and model details, we selected them as follows. We used the logistic activation function for the hidden layer, and a linear output layer. Training is performed for 500 epochs, using a momentum term 0.2, and an adaptive learning rate with initial value 0.01, an increase step of 1.05 and a decrease step of 0.7. The reason for choosing these values for the learning rate and momentum is that we found them in another large scale study (Ahmed et al [2]) to be quite effective when applying neural networks to time series forecasting.

7.3 The Gaussian Process Regression model (GPR)

Gaussian process regression (Rasmussen and Williams [48]) is a nonparametric regression model based on the concept of spatial smoothness. Specifically, if the training data are modeled as points in the input space, then the underlying function values for two adjacent points (adjacent in the input space) are assumed to be close to each other, or highly positively correlated. On the other hand, the further the points get, the looser the correlation becomes. This smoothness aspect is enforced using a Bayesian prior, as described below.

Let x_i be the input vector for training data point i , with corresponding observed response (or target output) y_i . Arrange the vectors x_i in a matrix X , and arrange the y_i 's in a vector y . The observed response y_i equals the underlying function value f_i plus some random error term ϵ_i (assume it is zero-mean normal with variance equal to σ_n^2). The function values f_i are smoothed versions of the observed responses y_i and are the inherent function values to be estimated. Also arrange the f_i 's in a vector f .

Some multivariate normal prior is attached to the vector f to enforce smoothness:

$$f \sim \mathcal{N}(0, V(X, X)) \quad (6)$$

where $V(X, X)$ denotes the covariance matrix between the function values of the different training data points, with the $(i, j)^{th}$ element of $V(X, X)$ being $V(x_i, x_j)$, the covariance between the function values of training data points i and j . To guarantee smoothness, the covariance function is taken as a monotonically decreasing function g of the distance between x_i and x_j :

$$V(x_i, x_j) = g(\|x_i - x_j\|^2) \quad (7)$$

For example, the exponential function is a typical choice for g , giving the following

form for the covariance matrix:

$$V(x_i, x_j) = \sigma_f^2 e^{-\frac{\|x_i - x_j\|^2}{2\alpha^2}} \quad (8)$$

The role of the prior is to impose smoothness of the solution (very smooth solutions are favored by the selected prior). Then, the observed responses are factored in, to obtain the posterior estimate of the function values. Given an input vector x_* , the prediction y_* can be derived using some Bayes rule manipulations as:

$$\hat{y}_* = V(x_*, X)[V(X, X) + \sigma_n^2 I]^{-1} y \quad (9)$$

For Gaussian process regression we have three key parameters that control the function smoothness: σ_n (the standard deviation of the error terms ϵ_i , see above), σ_f and the α (the parameters that pertain to the covariance function as in Eq. 8). We used the model selection algorithm proposed by Rasmussen and Williams [48]. It is an algorithm that maximizes the marginal likelihood function. More details and description of this algorithm can be found in [48].

Unlike neural networks, Gaussian process regression has received very little attention from the forecasting community. The largest study to-date is the comparative study by Ahmed et al [2], whereby Gaussian process regression occupied the second rank among a large group of computational intelligence models. Among the few other studies on using Gaussian process regression for time series forecasting are the works of Brahim-Belhouari and Bermak [11], Chapados and Bengio [12], and Girard et al [26].

8 Analysis

To shed some light and single out some lessons to be learned from this work, we attempted to find out what aspects of this forecasting system that are responsible for the superior rank. There are three major components of the proposed system: The concept of forecast combination, the novel deseasonalization method (using the median instead of the average), and the way the individual forecasts and the preprocessing are tested and selected. We considered only the first two aspects. It was hard to test the third aspect, because it is not clear what alternative to compare against (this means if we had not used this selection approach what other approach to use).

For the first concept we performed the following simple test. We computed the forecast error (the SMAPE) for each of the nine individual forecasts, and compared it with the forecast error of the combined forecast. This test will tell us whether or not the forecast combination aspect has had a decisive positive impact on the performance. We performed a multiple time origin test (see Tashman [52]). The time

origin denotes the point from which the (multi-step ahead) forecasts are generated. In the multiple time origin test we shift the time origin a few times, each time performing the forecast and computing the error. The average of these errors will then be used as the evaluation criterion. We used a three time origin test (each is one week apart) with the forecast periods being:

1. Day 680 to Day 735 (or Week 98 to Week 105).
2. Day 687 to Day 735 (or Week 99 to Week 105).
3. Day 694 to Day 735 (or Week 100 to Week 105).

In addition to the nine individual forecasting models, we also added a statistical benchmark for comparison, namely Holt’s additive exponential smoothing model. For this model we estimated the smoothing parameters by minimizing the sum of square errors on the training set, and estimated the initial level and trend by fitting a regression line on the first 50 data points and observing the intercept and slope, see Hyndman et al [38] (typically 5 or 10% of the training data points are used for obtaining these initial variables). The Holt’s model is applied on the deseasonalized time series (where we used the same seasonal decomposition approach as that used by the proposed forecasting models).

Table 1: The Performance Comparison of the Individual Forecasting Models Versus the Forecast Combination Model

Model	SMAPE % (Std Err)	Avg Rank	FRAC BEST (%)
GPR-ITER	19.90 (0.60)	6.34	7.21
GPR-DIR	21.22 (0.61)	8.41	3.60
GPR-LEV	20.19 (0.74)	5.74	12.61
NN-ITER	21.11 (0.70)	7.88	2.70
NN-LEV	19.83 (0.82)	4.96	27.93
MULT-REGR1	19.11 (0.60)	5.50	2.70
MULT-REGR2	18.96 (0.60)	4.96	9.01
MULT-REGR3	18.94 (0.60)	4.80	7.21
MOV-AVG	19.55 (0.61)	5.61	7.21
Holt’s Exp Sm	23.77 (1.13)	7.90	7.21
Combined	18.95 (0.76)	3.89	12.61

Table 1 shows some of the comparison results for the competing 11 models. The table shows the average SMAPE (for the multiple time origin test set) of each model

over all 111 time series. Moreover, it shows the average rank of each model, defined as follows. Consider time series i and let us rank all compared models (assume that we have k models, in our case $k = 11$), with the rank being 1 for the best model and k for the worst. The rank is on the basis of the accuracy (SMAPE-wise) in the multiple time origin test set. Let r_{ij} be the rank of model j on time series i . The average rank R_j of model j is the average of the ranks of model j over all the time series (let there be n time series, where $n = 111$ in our case). The average ranks R_j of the different models are shown in the third column in the table. We also computed another rank-based measure, namely the fraction best (or in short FRAC-BEST). It is defined as the fraction of time series for which a specific model beats all other models. We used the SMAPE as a basis for computing this measure. The reason why this measure could be of interest is that a model that has a high FRAC-BEST, even if it has average overall SMAPE, is deemed worth testing for a new problem, as it has a shot at being the best. We can see from the table that the forecast combination model outperforms the majority of the individual forecasting models and also the statistical benchmark SMAPE-wise. (Only MULT-REGR2 and MULT-REGR3 are about equal in performance to the combined model, while the remaining models are lower in performance). Concerning the fraction best measure, NN-LEV is far ahead of the pack and GPR-LEV gives equal performance (to the combined model). We also see that the average rank of the forecast combination model is 3.89, beating all other ten models. To test whether this outperformance is statistically significant, we designed the following test.

The goal here is to test if the accuracy of the considered combined forecasting model is significantly different from that of each of the nine constituent forecasting models and the statistical benchmark. These tests call into question the problem of comparing multiple models on multiple data sets, and trying to infer if there are significant general differences in performance. For such case Demsar [19] in a detailed comparative study recommends using a two stage procedure: first to apply Friedman’s test to test if the compared models have the same mean rank. If this test rejects the null-hypothesis, then post-hoc pairwise tests are to be performed to compare the different models. These tests adjust the critical values higher to ensure that there is at most a 5% chance that one of the pairwise differences will be erroneously found significant.

The Friedman’s test is a nonparametric test, designed to detect differences among two or more groups. Friedman’s test, operating on the mean ranks R_j , considers as the null hypothesis that all models are equivalent in performance (have similar mean ranks). Under the null hypothesis the following statistic:

$$\chi_F^2 = \frac{12n}{k(k+1)} \left[\sum_{j=1}^k R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (10)$$

is approximately distributed as χ^2 with $k - 1$ degrees of freedom.

If the Friedman’s test is rejected we proceed with the post-hoc tests. We consider the following statistic for the pairwise tests between model j and model l :

$$D = (R_j - R_l) / \sqrt{\frac{k(k+1)}{6n}} \quad (11)$$

Under the null hypothesis of equal mean ranks this statistic is approximately normal. In multiple comparisons, because of the possibly large numbers of pairwise comparisons, there is a relatively high chance that some pairwise tests are incorrectly rejected. There are a number of methods that adjust the critical value of the test to account for this bias. When the tested models are compared to a control model, then the most suitable approach is to use Bonferroni-type corrections. (This applies to our situation where we are comparing the combined forecasting model with each of the constituent forecasting models.) The basic method, the Bonferroni-Dunn test (see Demsar [19]), is known to be too conservative. More powerful tests have been developed in the literature, such as Holm’s step-down procedure [35], Hochberg’s step-up procedure [32], and Hommel’s procedure [36]. The latter two are the more powerful of these three tests (see Hochberg [32] and Demsar [19]), so we used Hochberg’s test. In this test we sort the p-values of the statistic D (Eq. 11) for the $k - 1$ comparisons with the control model (i.e. the forecast combination model). The largest p-value is compared with the critical value ($\alpha = 0.05$), then the next largest is compared with $\alpha/2$, then the next with $\alpha/3$, and so on, until it encounters a hypothesis that it can reject. When that happens, all hypotheses with smaller p-value are rejected as well.

Performing the comparison between the forecast combination model, its nine constituent models, and the statistical benchmark, and applying the Friedman’s test, we found that the null hypothesis is rejected at the 5% level (the statistic $\chi^2_F = 218.3$), indicating significant differences in the mean ranks among the 11 compared models. Subsequently, we applied Hochberg’s test, and the results indicate that the highest p-value of the pairwise comparisons with the forecast combination model is that with MULT-REGR3, giving a p-value of 0.041. Since it is less than 0.05, we reject all hypotheses (at the 5% level), indicating a significant difference in mean rank between the forecast combination model and each of the ten other competing models.

Commenting on the results, we observe that for the case of the mean rank, confirming our Friedman/Hochberg tests, the combined model achieves best performance. Even if we focus on the SMAPE measure, it is a good achievement that the combined forecasting model beats the majority of the constituent models and equals in performance the best performing constituent model (in that measure). The reason is that a priori, we do not know which model will turn out to be the best, especially with all the changing conditions that time series typically pass through. The curious

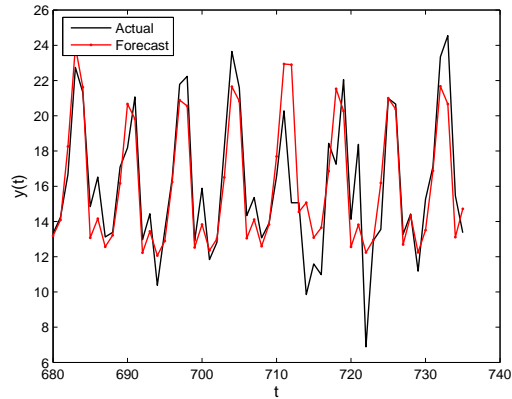


Figure 6: The forecasts of the developed model for Time Series No. 20 versus the actual for the validation period.

observation is that NN-LEV was considerably better than the other models in the fraction best model. This indicates that on average this model tends to perform better than the others, but once in a while it completely fails. (The reason it fails could be due to the occurrence of overfitting, or due to the failing of the training algorithm to reach any acceptable solution.) This perhaps suggests an approach whereby we disable using this model (based on some training set or validation set error) if it is deemed to perform poorly. But this will not be tested here and could be investigated in a future study. To illustrate the forecasting ability of the combined system, Figure 6 shows the forecast versus the actual for the test period of one of the time series.

Table 2: The Performance Comparison of the Forecasting Model Using Median Based Deseasonalization Versus that Using Multiplicative Seasonal Decomposition

Model	SMAPE % (Std Err)	Avg Rank
AVG SEAS DECOMP	19.56 (0.60)	1.89
MEDIAN SEAS	18.95 (0.76)	1.11

Concerning the deseasonalization issue, we tested the used median-based deseasonalization against the well-known multiplicative seasonal decomposition method (see Makridakis et al [42] and Gardner and Diaz-Saiz [23]). This method is widely used in the forecasting literature and therefore it is reasonable to use it as a benchmark. We considered the forecasting system that we have developed, and replaced all

median-based deseasonalization steps with the multiplicative seasonal decomposition (leaving everything else the same), to check on the performance of the alternative hypothesis. Table 2 shows the SMAPE’s and the average ranks of the forecasting system with the proposed median deseasonalization method and the forecasting system with the multiplicative seasonal decomposition method. Again one can see that the proposed deseasonalization method leads to better performance in comparison to the alternative seasonal decomposition method. Friedman’s test, applied on the ranks, indicates that the mean rank difference is significant at the 5% level.

We wish to mention that when we had developed the forecasting system we did not overuse the test set for the forecast combination and the seasonality aspects. Most of the experimentation on the test set was for the purpose of selecting the best nine individual models. But, we had determined at the outset to use the forecast combination concept. Concerning deseasonalization, we used only one test to select (to compare the proposed deseasonalization approach with the multiplicative seasonal decomposition method) and there was no extensive experimentation. So the test set is not polluted with multiple experimentation and model selection (with respect to the forecast combination and seasonality aspects) that tends to bias the test results.

9 Conclusions and Discussion

In this paper we have presented the model with which we participated in the NN5 time series competition. The model obtained a high rank, and this is mainly because of several aspects. We have shown that the forecast combination aspect was one of the most significant. Also, the median-based deseasonalization method clearly added value. We also believe that the careful and principled selection of the individual forecasting models and the preprocessing methods that we followed had a considerable role in leading to improved results.

However, the contribution of this last point (i.e. the individual selection of models) could not be quantified. In our proposed approach, we considered 140 models/preprocessing combinations, and selected the best 9 to combine. The question is whether we have tested enough models. Perhaps testing too many would lead to spurious final selections (especially with highly specified nonlinear models). Perhaps more targeted choices of the types of tested models would be beneficial. Even though it is very hard to know a priori which model is expected to work, some general guidelines would yield some needed focus in the search.

But, what we would like to champion is the value of using a validation or a test period (like the multiple time origin test). We observed that a parameter set or model favored by the test set, even if this goes against some intuitive arguments, generally prevails in true out of sample performance. We have not performed a quantitative

study for the value of the test set, but some other studies have analyzed this problem (for example [3]). A very promising future research direction is to develop more efficient validation procedures, for the purpose of assessing and selecting the right models. These could borrow some of the specialized techniques found in some other related problems, such as in classification error estimation, where state of the art methods based on bootstrap sampling have proven superiority.

Acknowledgement

The authors would like to acknowledge the help of Nesreen Ahmed, who has helped in developing some of the programs used. The authors would like to acknowledge the fruitful discussions with Sherif Hashem of Cairo University. This work was supported by the *Data Mining for Improving Tourism Revenue in Egypt* research project within the Egyptian Data Mining and Computer Modeling Center of Excellence.

References

- [1] M. Adya and F. Collopy, *How effective are neural networks at forecasting and prediction? A review and evaluation*, Journal of Forecasting, 17, 481-495, 1998.
- [2] N. K. Ahmed, A. F. Atiya, N. El Gayar, and H. El-Shishiny, *An empirical comparison of machine learning models for time series forecasting*, Accepted in the Special Issue on *The Link Between Statistical Learning Theory and Econometrics: Applications in Economics, Finance, and Marketing*, Econometric Reviews, to appear 2010.
- [3] M. Aiolfi and A. Timmermann, *Persistence in forecasting performance and conditional combination strategies*. Journal of Econometrics 135, 31-53, 2006.
- [4] I. Alon, M. Qi, and R. J. Sadowski, *Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods*, Journal of Retailing and Consumer Services, 8, 147-156, 2001
- [5] R. Andrawis and A. F. Atiya, *A new Bayesian formulation for Holt's exponential smoothing*, Journal of Forecasting, 28(3), 218-234, 2009.
- [6] J. S. Armstrong, *Combining forecasts*, in Principles of Forecasting: A Handbook for Researchers and Practitioners, J. S. Armstrong (ed.), Norwell, MA: Kluwer Academic Publishers, 2001.

- [7] A. F. Atiya, S. M. El-Shoura, S. I. Shaheen, and M. S. El-Sherif, *A comparison between neural network forecasting techniques - case study: river flow forecasting*, IEEE Transactions Neural Networks, 10(2), 402-409, 1999.
- [8] S.D. Balkin and J.K. Ord, *Automatic neural network modeling for univariate timeseries*, International Journal of Forecasting, 16(4), 509-15, 2000.
- [9] J. M. Bates, and C.W.J. Granger, *The combination of forecasts*, Operations Research Quarterly, 20, 451-468, 1969.
- [10] L. J. Callen, C. C.Y. Kwan, P. C. Y. Yip, and Y. Yuan, *Neural network forecasting of quarterly accounting earnings*, International Journal of Forecasting, 12, 475-482, 1996.
- [11] S. Brahim-Belhouari and A. Bermak, *Gaussian process for nonstationary time series prediction*, Computational Statistics & Data Analysis, 47, 705712, 2004.
- [12] N. Chapados and Y. Bengio. *Augmented functional time series representation and forecasting with Gaussian processes*, in B. Schölkopf, J. C. Platt, and T. Hoffman, editors, Advances in Neural Information Processing Systems 19, pages 457-464, Cambridge, MA, U.S.A., 2007. The MIT Press.
- [13] H. Chen and J. E. Boylan, *Empirical evidence on individual, group and shrinkage seasonal indices*, International Journal of Forecasting, 24, 525534, 2008.
- [14] R. T. Clemen, *Combining forecasts: A review and annotated bibliography*, International Journal of Forecasting, 5, 559-583, 1989.
- [15] S. Crone, *Results of the NN5 time series forecasting competition*, Presentation at the IEEE World Congress on Computational Intelligence, WCCI'2008, Hong Kong, June 2008.
- [16] W. L. Crum, *The use of median in determining seasonal variation*, Journal of the American Statistical Association, 18, 607-614, 1923.
- [17] L. M. de Menezes, D. W. Bunn, and J. W. Taylor, *Review of guidelines for the use of combined forecasts*, European Journal of Operational Research, 120, 190-204, 2000.
- [18] T. Hill, M. OConnor, and W. Remus, *Neural network models for time series forecasts*, Management Science, 42, 1082-1092, 1996.
- [19] J. Demsar. *Statistical comparisons of classifiers over multiple data sets*, Journal of Machine Learning Research, 7, 130, 2006.

- [20] F. X. Diebold and J. A. Lopez, *Forecast evaluation and combination*, in G. S. Maddala, C. R. Ran, Eds, *Statistical Methods in Finance*, Handbook of Statistics, Vol. 14, Elsevier, Amsterdam.
- [21] T. G. Dietterich, *Ensemble methods in machine learning*, Proceedings of the First International Workshop on Multiple Classifier Systems, 1-15, 2000.
- [22] M. Friedman, *A comparison of alternative tests of significance for the problem of m rankings*, *Annals of Mathematical Statistics*, 11, 8692, 1940.
- [23] E. S. Gardner and J. Diaz-Saiz, *Seasonal adjustment of inventory demand series: a case study*, *International Journal of Forecasting*, 18, 117-123, 2002.
- [24] E. Gardner, "Exponential smoothing: The state of the art—Part II", *International Journal of Forecasting*, Vol. 22, pp. 637-666, 2006.
- [25] E. Ghysels, D. Osborn, and P. Rodrigues, *Forecasting seasonal rime series*, in G. Elliott, C. W. J. Granger, and A. Timmermann, Eds. *Handbook of Economic Forecasting*, Elsevier Pub., 659-711, 2006.
- [26] A. Girard, C. Rasmussen, J. Quinonero-Candela, and R. Murray-Smith. *Multiple-step ahead prediction for non linear dynamic systems - a Gaussian process treatment with propagation of the uncertainty*, In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2003. The MIT Press.
- [27] V. Gomez and A. Maravall, *Programs TRAMO and SEATS, instructions for the user (beta version: September 1996)*, Working Paper 9628, Bank of Spain, 1996.
- [28] C. W. J. Granger and Y. Jeon, *Thick modeling*, *Economic Modeling*, 21(2), 323-343, 2004.
- [29] S. Hashem, *Optimal Linear Combinations of Neural Networks*, Ph.D. Thesis, Purdue University, 1993
- [30] S. Hashem and B. Schmeiser, *Improving model accuracy using optimal linear combinations of trained neural networks*, *IEEE Transactions on Neural Networks*, 6(3), 792-794, 1995.
- [31] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics. Springer-Verlag, 2001.
- [32] Y. Hochberg, *A sharper Bonferroni procedure for multiple tests of significance*, *Biometrika*, 75, 800-803, 1988.

- [33] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, *Bayesian model averaging: a tutorial*, *Statistical Science*, 14(4), 382-417, 1999.
- [34] M. Hollander, and D. A. Wolfe, *Nonparametric Statistical Methods*, Wiley, 1973.
- [35] S. Holm, *A simple sequentially rejective multiple test procedure*, *Scandinavian Journal of Statistics*, 6, 6570, 1979.
- [36] G. Hommel, *A stagewise rejective multiple test procedure based on a modified Bonferroni test*, *Biometrika*, 75, 383386, 1988.
- [37] S. Hylleberg, *Modelling Seasonality*, Oxford University Press, Oxford, UK, 1992.
- [38] R. Hyndman, A. Koehler, K. Ord, R. Snyder, and S. Grose, *A state space formulation for automatic forecasting using exponential smoothing methods*, *International Journal of Forecasting*, Vol. 18, pp. 439-454, 2002.
- [39] D. M. Kline, *Methods for multi-step time series forecasting with neural networks*, in *Neural Networks for Business Forecasting*, G. P. Zhang (ed.), Information Science Publishing, Hershey, PA, 226-250, 2004.
- [40] R. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*, *Proceedings International Joint Conference on Artificial Intelligence*, IJCAI, 1995.
- [41] S. Makridakis, A. Anderson, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, R. Winkler, *The accuracy of extrapolation (time series) methods: results of a forecasting competition*, *Journal of Forecasting*, 1, 111-153, 1982.
- [42] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting: Methods & Applications*, 3rd Edition, Ch. 3, Wiley, 1998.
- [43] S. Makridakis and M. Hibon, *The M3-competition: results, conclusions, and implications*, *International Journal of Forecasting*, 16, 451-476, 2000.
- [44] M. Nelson, T. Hill, W. Remus, and M. O'Connor, *Time series forecasting using neural networks: should the data be deseasonalized first?*, *Journal of Forecasting*, 18(5), 359-367, 1999.
- [45] D. Nguyen and B. Widrow, *Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights*, *Proceedings of the International Joint Conference on Neural Networks*, 3, 21-26, 1990.

- [46] NN5, *Forecasting Competition for Artificial Neural Networks & Computational Intelligence*, 2008, <http://www.neural-forecasting-competition.com/NN5/results.htm>.
- [47] M. Qi and G. P. Zhang, *Trend timeseries modeling and forecasting with neural networks*, IEEE Transactions on Neural Networks, 19(5), 808-816, 2008.
- [48] C. E. Rasmussen, and C. K. L. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
- [49] D. J. Ried, *A Comparative Study of Time Series Prediction Techniques on Economic Data*, Ph.D. Thesis, University of Nottingham, Nottingham, UK, 1969.
- [50] R. Sharda and R.B. Patil, *Connectionist approach to time series prediction: An empirical test*, Journal of Intelligent Manufacturing, 3, 317-323, 1992.
- [51] J. H. Stock and M. Watson, *Combination forecasts of output growth in a seven-country data set*, Journal of Forecasting, 23, 405-430, 2004.
- [52] L. Tashman, *Out-of-sample tests of forecasting accuracy an analysis and review*, International Journal of Forecasting, 16, 437-450, 2000.
- [53] T. Terasvirta, D. van Dijk, and M. C. Medeiros, *Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A reexamination*, International Journal of Forecasting, 21, 755-774, 2005.
- [54] A. Timmermann, *Forecast combinations*, in G. Elliott, C. W. J. Granger, and A. Timmermann, Eds. *Handbook of Economic Forecasting*, Elsevier Pub., 135-196, 2006.
- [55] S. Wheelwright and S. Makridakis, *Forecasting Methods of Management*, Wiley, New York, 1985.
- [56] G. P. Zhang and M. Qi, *Neural network forecasting for seasonal and trend time series*, European Journal of Operational Research, 160, 501-514, 2005.
- [57] G. P. Zhang and D. M. Kline, *Quarterly time-series forecasting with neural networks*, IEEE Transactions on Neural Networks, 18(6), 1800-1814, 2007.