Artif Intell Rev (2019) 52:2019–2037
https://doi.org/10.1007/s10462-017-9593-z

A methodology for applying *k*-nearest neighbor to time series forecasting

Francisco Martínez[1] · **María Pilar Frías**[2] · **María Dolores Pérez**[1] · **Antonio Jesús Rivera**[1]

Published online: 21 November 2017

Abstract In this paper a methodology for applying *k*-nearest neighbor regression on a time series forecasting context is developed. The goal is to devise an automatic tool, i.e., a tool that can work without human intervention; furthermore, the methodology should be effective and efficient, so that it can be applied to accurately forecast a great number of time series. In order to be incorporated into our methodology, several modeling and preprocessing techniques are analyzed and assessed using the N3 competition data set. One interesting feature of the proposed methodology is that it resolves the selection of important modeling parameters, such as *k* or the input variables, combining several models with different parameters. In spite of the simplicity of *k*-NN regression, our methodology seems to be quite effective.

Keywords Nearest neighbors · Time series forecasting · Combined forecast · Feature selection

1 Introduction

Time series forecasting has been performed traditionally using statistically based methods such as Box and Jenkins methodology for ARIMA models (Box et al. 2008) or exponential smoothing techniques. Over time, new tools and developments have improved these methodologies, e.g. the use of information criteria for model selection or the discovery of state space models to give a sound statistical foundation to exponential smoothing heuristics (Hyndman et al. 2008). On the other hand, the last decades have witnessed the widespread use of computational intelligence approaches to forecast time series. International journals

This paper has been partially supported by the project TIN2015-68854-R (FEDER Founds) of the Spanish Ministry of Economy and Competitiveness.

✉ Francisco Martínez
fmartin@ujaen.es

1 Computer Science Department, University of Jaén, Campus Las Lagunillas s/n, 23071 Jaén, Spain

2 Statistics and Operations Research Department, University of Jaén, Jaén, Spain

and conferences have published thousands of publications on forecasting with computational intelligence methods; however, this field has not achieved yet the degree of maturity of statistical methodologies.

A natural question arises about whether computational intelligence methods outperform traditional statistical models. The NN3 competition (Crone et al. 2011) tried to shed some light on this question comparing the performance of artificial intelligence techniques against statistical models after the modest result of the only artificial neural network contender in the M3 competition (Makridakis and Hibon 2000). One of the contenders of the NN3 competition used a k-NN regression method with very good results. This is a bit surprising, given the simplicity of k-NN. Unfortunately, the methodology used to apply this method is not explained anywhere, apart from an incomplete outline in the competition website.[1]

Motivated by the lack of a methodology for applying k-NN regression to time series forecasting in a general setting, this paper analyzes different preprocessing and modeling strategies in order to get some insights about the impact of these strategies on the performance of a k-NN regression algorithm. We think that many of the strategies used in the proposed methodology are general enough to be applied in other time series forecasting approaches based on computational intelligence techniques.

The remainder of this paper is structured as follows. Section 2 explains how k-NN regression can be applied to forecast time series and some related work. Section 3 describes the requirements of the methodology we want to develop. Sections 4 and 5 describe different preprocessing and modeling techniques that can be applied to improve forecasting accuracy on a k-NN regression setting. Section 6 experiments with the effectiveness of the different preprocessing and modeling techniques explained in previous sections. In Sect. 7 the chosen methodology is presented and its forecast accuracy assessed. Finally, Sect. 8 draws some conclusions.

2 Forecasting time series with k-NN

The k-nearest neighbors is a well-known nonparametric method used for classification and regression. It has been labeled as a lazy learner as the training step consists in storing the instances verbatim. Given some features—explanatory variables—of a new instance to be classified—regressed on—k-NN finds the k training instances that are closest to the new instance according to some distance metric and returns their majority class—average explained variable.

In order to apply k-NN regression on an univariate time series forecasting setting the explanatory variables are lagged values of the explained or forecast variable. Figure 1 shows an example of one-step-ahead forecasting using lags 1–3 as explanatory variables. The last three values of the time series are the new instance to be regressed on and the two sets of consecutive black dots the 2 nearest neighbors, whose targets are the triangles that are averaged to produce the forecast—the asterisk. The underlying intuition to using k-NN on time series forecasting is that any time series contains repetitive patterns, so we can find previous similar patterns to the current series structure and use their subsequent patterns to predict the future behavior.

Despite early work (Yakowitz 1987), it is difficult to find applications of k-NN regression to time series forecasting. In Ahmed et al. (2010), a comparison of several machine learning algorithms—including k-NN—applied in time series forecasting is conducted. Unfortunately,

[1] http://www.neural-forecasting-competition.com/NN3/index.htm.

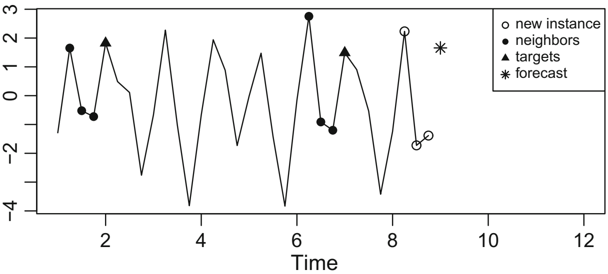

Fig. 1 One-step-ahead forecast with 2-NN regression

only one-step-ahead predictions are considered and there is no information about how important modeling parameters, such as the input variables, are selected. In this study multilayer perceptron an Gaussian processes achieve the best results. In the field of electricity load and price forecasting some authors have used *k*-NN regression, but the methodology is adjusted to the specific problem; for example, in Lora et al. (2003), the next day hourly load is predicted selecting always as input variables the load of the 24 h of the previous day. In the area of finance, Fernandez-Rodriguez et al. (1999) uses *k*-NN as a tool for forecasting exchange-rate. In this case, the nearest neighbors are used to do linear autoregressive predictions, an approach quite different to the one proposed in this paper. In Zhang et al. (2017), a multidimensional *k*-NN model combined with a new proposed ensemble empirical mode decomposition is used to forecast stock prices. A similar approach is used in Ren and Suganthan (2014), for wind speed forecasting.

3 Goal of this work

The goal of this work is to develop an automatic methodology for applying *k*-NN regression to forecast time series in an effective and efficient way. This methodology comprises a set of preprocessing and modeling tasks that are described in subsequent sections. To assess the degree of effectiveness of our methodology we have compared its forecast accuracy against the contenders of the NN3 competition, specially against the *k*-NN model submitted to that competition.

Some tools used in our methodology are applied conditionally based on their results on an validation set using a rolling origin evaluation. The rolling origin evaluation (Tashman 2000) is a technique for assessing forecast accuracy that tries to get the most of the out-of-sample or test data. In order to do so, the origin of forecasts advances one observation at a time and different models are fitted, and forecasts generated, for the different origins. For example, if the test data comprises 6 observations we would have an initial set of forecasts—and errors—for horizons 1, 2, …, 6. Then, the forecast origin moves one step forward, another model is fitted and a set of forecasts for horizons 1, 2, …, 5 are generated with the new model. This process is repeated, so that we can assess 6 one-step-ahead predictions, 5 two-step-ahead predictions and so on. This is a great contrast with the fixed origin evaluation that, for a test data of size 6, is only able to assess 1 prediction for each one of the 6 horizons.

Of course, the rolling origin evaluation is computationally intensive because it has to fit as many models as observations in the test data. An intermediate approach is to reduce the number of origins. For example, if the test data comprises 6 observations we might use two

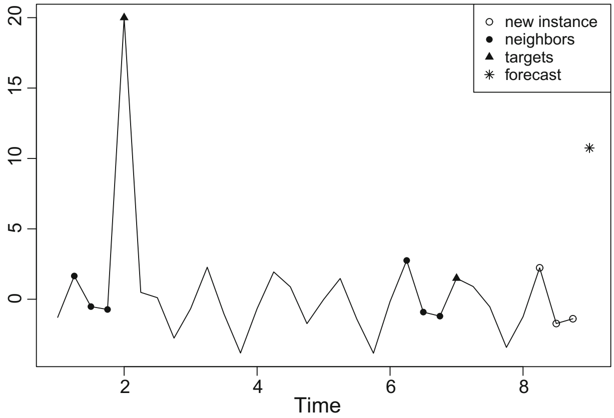

Fig. 2 Outlier with 2-NN regression

origins: one at the last observation of the training data and the other at the third observation of the test data. This way, only two models are fitted and we can assess two predictions for horizons 1–3 and one prediction for horizons 4–6.

It is worth noting that we want to develop an automatic methodology, so that time series can be forecast without human intervention. Therefore, the preprocessing and modeling strategies analyzed in the following sections should satisfy this requirement.

4 Preprocessing

When a time series does not meet some properties as stationarity or additive seasonality, classical time series forecasting methodologies usually transform the original series to obtain the aforementioned properties. This way, a model that fits better the transformed series can be found. In this section we have analyzed different preprocessing or transformation strategies that can be applied to improve the forecast accuracy of a k-NN regression model used in a time series forecasting context.

4.1 Normalization

Normalization is routinely applied to k-NN classifiers so that the distance metric is not biased towards attributes that have larger scales of measurement (Witten et al. 2011). However, as all the observations of a time series are in the same scale we consider that normalization is not necessary for doing k-NN regression on a time series.

4.2 Outliers

An outlier is an extreme observation that, if not adjusted, may cause serious estimation and forecast errors (Ord and Fildes 2003) . It can be associated to a real observation, but its value is so atypical that can distort the forecasts. For example, in Fig. 2 the target of the first of the 2-nearest neighbors is an outlier, distorting the forecast.

An extreme observation should be adjusted by an expert after careful thought depending on whether the extreme phenomenon can happen again in the future. However, an automatic

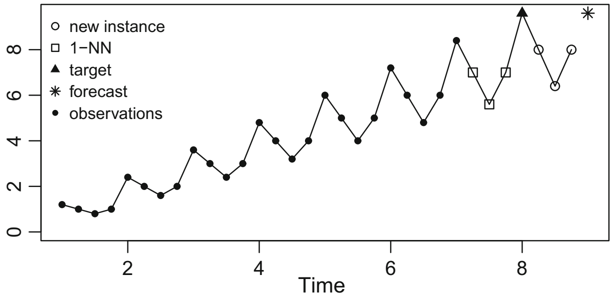

Fig. 3 One-step-ahead forecast with 1-NN regression of a series with multiplicative seasonality

forecasting tool has no room for any expert intervention. We have adopted the heuristic described in Yan (2012), to detect and adjust outliers, as our experiments have confirmed its usefulness. In that paper an observation is deemed an outlier if its absolute value is four times greater than the absolute medians of the three consecutive points before and after the observation.

$$|o_i| \geq 4 \times max\{|m_b|, |m_a|\}$$

where $m_b = median(o_{i-3}, o_{i-2}, o_{i-1})$ and $m_a = median(o_{i+1}, o_{i+2}, o_{i+3})$. When an observation is considered an outlier its value is replaced by the average value of the two observations that are immediately before and after the outlier.

4.3 Box–Cox transformations

Box–Cox transformations (Hyndman and Athanasopoulos 2014) are defined as follows:

$$w_i = \begin{cases} \log(x_i) & \text{if } \lambda = 0; \\ (x_i^\lambda - 1)/\lambda & \text{otherwise.} \end{cases}$$

The Box–Cox transformation is used, for instance, in ARIMA modeling to stabilize the variance in series with multiplicative seasonality. These series are also problematic for k-NN regression as the seasonal component is increasing over time. Since k-NN regression predicts average target values in the historical data, it is not able to predict values that lie outside the range of the historical observations. Figure 3 shows an example of an artificial time series with multiplicative seasonality and an increasing trend. The one-step-ahead forecast generated by 1-NN is the maximum value in the historical data. However, this forecast will undershoot the actual value as long as the increasing trend and multiplicative seasonality pattern is repeated into the future.

4.4 Trend

When a time series has a global trend, the likelihood of a future value being out of the range of the historical data is very high—that is, the same problem as with multiplicative seasonality. Therefore, k-NN is not suitable for forecasting time series with a global trend and some tools are needed to detect and remove the trend component. In this section we explain some strategies for detrending a time series.

The first strategy is differencing. In the ARIMA methodology (Hyndman and Athanasopoulos 2014), differences at lag 1 are commonly used to stabilize the mean level of a time series with a trend. In a k-NN regression setting, if the lags used as input variables are the set l, then differences at lag $min(l)$ should be taken.

Another detrending approach is based on doing an STL decomposition (Cleveland et al. 1990). When this approach is used, the series is decomposed into a trend component and a seasonal and remainder component. The trend component is forecast by using some suitable method, as Holt's method, and the seasonal and remainder component is forecast using k-NN. The two forecasts are added—assuming an additive decomposition—to generate the final forecast.

4.5 Seasonality

Although there is no reason to think that k-NN cannot model additive seasonality, we have experimented with two strategies for removing seasonality to assess whether they can improve the forecast accuracy of a k-NN model.

The first one is very well-known since it is part of the ARIMA methodology. It consists in taking differences at lag n, where n is the period of seasonality. The second one is described in Yan (2012). In this work an heuristic test is done for detecting seasonality; if detected, deseasonalizing is performed by subtracting the seasonal average. For example, for monthly data all the January observations are subtracted by the average January value. Forecasts are generated on the seasonally adjusted series and the seasonal averages are added back to these forecasts to get the final predictions.

5 k-NN modeling

To forecast time series using k-NN some decisions have to be made about the value of k, the selection of input variables, the multi-step ahead strategy, the distance metric and the way targets are combined. In the following subsections we explain different alternatives for these modeling decisions.

5.1 Choosing k

There are two common approaches to selecting the number of neighbors, i.e., k. A first, fast, straightforward solution is to use some heuristic. For example, it is recommended setting k to the square root of the number of training instances. The other approach is to select k using an optimization tool. In this case, the training set is further divided into a training and a validation set and k is selected so that it minimizes a forecast accuracy measure for the validation data using the training data. The rolling origin method explained in Sect. 3 has been used to assess forecast accuracy. A disadvantage of using an optimization tool is that it might be very time consuming.

We have also tested a third strategy that tries to balance the benefits of efficiency and an exhaustive search. This strategy is inspired by the success achieved by model combination in time series forecasting. Model combination has a long history in time series forecasting (Bates and Granger 1969; Hibon and Evgeniou 2005) and one of the four conclusions of the M3 competition (Makridakis and Hibon 2000) is that "the accuracy when various methods are being combined outperforms, on average, the individual methods being combined". The third strategy simply combines the forecasts of several k-NN models with different preset k values, generating as final forecast the average forecast of the different models. This way, the use of a time consuming optimization tool is avoided and the forecasts are not based on an unique, heuristic k value.

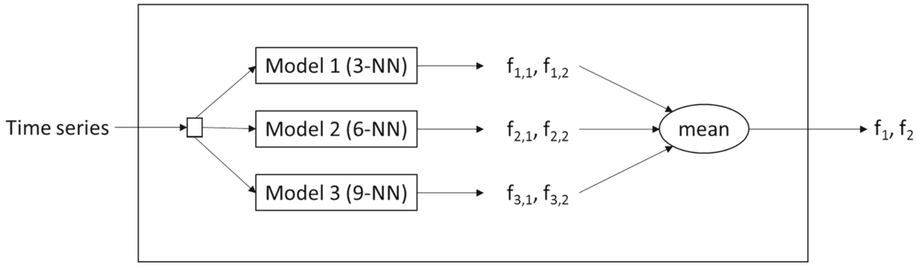

Fig. 4 Third strategy for selecting k: combining models with different k values

Next, the third strategy for selecting k is explained in more detail. Let us suppose that a combination of three k-NN models is used. The preset k values of the models are 3, 6 and 9 respectively. Let us also suppose that the next two future points are to be forecast. Figure 4 illustrates how this strategy works. Given the time series, the three models are trained, each one with a different k value. Each model generates its forecasts—because it is a two-steps ahead forecast, the models will use some multi-step ahead strategy, see Sect. 5.3. The forecasts of the models are averaged to generate the combined forecast, which is the final forecast produced by the combination model. In Fig. 4, f_1 is the mean of the forecasts of the three models for horizon 1, that is, $\frac{f_{1,1}+f_{2,1}+f_{3,1}}{3}$ and f_2 is the mean of the forecasts for horizon 2: $\frac{f_{1,2}+f_{2,2}+f_{3,2}}{3}$.

5.2 Choosing the input variables

In this section we explain several strategies for selecting the lagged observations of a time series used as input variables for the k-NN regression model. A first, naive approach is to use a default set of lags. For example, lags 1 to n could be used, where n is the period of seasonality, i.e., lags 1–12 for monthly series, lags 1–4 for quarterly series and so on.

A second approach is to use the partial autocorrelation function (PACF) to select as input variables the lags with significant autocorrelation. Although the PACF only tests for linear relationship, experience has shown us that this is an effective way of selecting input variables. This can be considered as a kind of filter approach for attribute selection (Freitas 2002).

The alternative to filtering approach is the wrapper approach. In the wrapper approach the input variables are selected taking into account their predictive performance on a validation set using the forecast algorithm. In our case the algorithm is k-NN regression and the rolling origin technique has been used to assess forecast performance. In order to avoid a time consuming exhaustive search on the input variables, some techniques as forward selection, backward elimination or forward-backward selection can be used (Freitas 2002).

5.3 Selecting a multi-step ahead forecasting approach

Sometimes, all that is needed is to forecast the next time point into the future. However, it is very common that forecasts are needed for several time points or horizons into the future. In fact, multi-step ahead predictions is one of the challenges that time series forecasting imposes on traditional regression models as k-NN.

Classical time series forecasting methodologies, as ARIMA or exponential smoothing, use the so-called recursive approach to implement multi-step ahead forecasting. In this approach the forecast function generates only one-step-ahead forecasts, which are used as input vari-

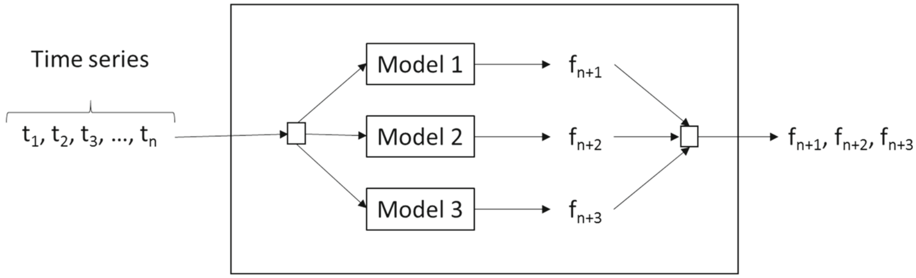

Fig. 5 The direct strategy for three-steps ahead forecasting

ables when historical observations are not available. For example, if lags 1–2 are used as input variables and the forecast horizon is 3, then the forecast for the first future point is generated by using the last two historical observations as input variables. The input variables used to forecast the second future point are the last historical observation and the previous forecast. Finally, the forecast of the third future point is based on the two previous forecasts. The obvious shortcoming of the recursive approach is the cumulative impact of forecast errors on future predictions. Anyway, some of the arguably most successful methods for time series forecasting are based on this approach.

On the other hand, the direct approach uses only historical data to predict the future. In this approach a different and independent model is fitted for forecasting every future point. For example—see Fig. 5—, if the forecast horizon is 3 three different models are fitted. The model forecasting the first future point—t_{n+1}—can use any lag as input variables. However, the model forecasting the second future point—t_{n+2}—must use as input variables lags equal or greater than 2. This is due to the fact that t_{n+1} is unknown and therefore cannot be used in the vector to be regressed on. For the same reason, the model forcasting the third future point must use as input variables lags equal or greater than 3—t_{n+1} and t_{n+2} are unknown. Of course, the direct approach is more time consuming since a different model has to be fitted for every future point. Furthermore, it ignores the dependencies of input variables. A third disadvantage of this approach is that the models for the further future points have fewer training instances. This can be very noticeable in short time series.

Although the recursive and direct approaches are the main strategies for multi-step ahead forecasting, other approaches exist. We have also experimented with the so-called Multi-Input Multi-Output (MIMO) strategy. The MIMO strategy learns one Multiple-Output model for a time series, that is, the model predicts all the future points at once—see Fig. 6. The MIMO strategy is usually applied in electricity load forecasting, when the hourly demand for the next day has to be forecast (Al-Qahtani and Crone 2013). Compared with the recursive strategy, MIMO reduces dramatically the number of training instances when the forecast horizon is high and the time series is short.

In Ben Taieb et al. (2012), a detailed description of these strategies for multi-step ahead forecasting can be found.

5.4 Distance metric and combination of targets

The distance metric plays a crucial role in a nearest neighbor algorithm, as it determines the way that the nearest neighbors are chosen. In classification, the Euclidean distance is by far the most used distance metric (Witten et al. 2011). All the papers in the references that apply

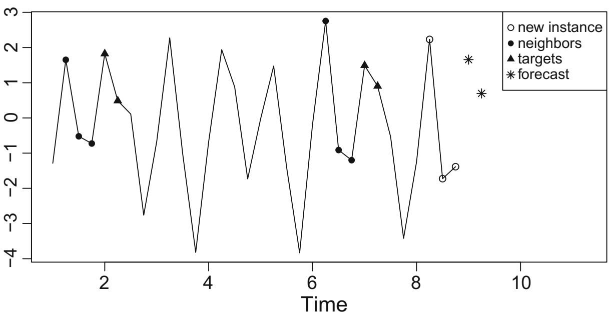

Fig. 6 Using 2-NN regression and the MIMO strategy for forecast horizon 2

k-NN to time series forecasting use the Euclidean distance as distance metric. Anyway, a comparison between the Euclidean and Manhattan distances is done in Sect. 6.1.2.

Given the k targets associated to the nearest neighbors, they have to be combined to produce the forecast. We have experimented with the following ways of combining the targets:

- The mean value.
- The median value.
- A weighted mean in which the weight of a target is related to the closeness of its associated neighbor to the instance to be forecast. Closer neighbors have higher weights.
- A trimmed mean in which the lowest and highest values are remove from the mean. If k is lower than or equal to 2 then the trimmed mean is computed as the mean.

6 Experimentation

The data from the NN3 competition (Crone et al. 2011) has been used as experimental data. This competition used 111 time series drawn from the M3 monthly industry data (Makridakis and Hibon 2000). The data set contains a balanced mix of 25 short-seasonal series, 25 long-seasonal series, 25 short-nonseasonal series and 25 long-nonseasonal series, apart from 11 experimental series. The short series have less than 52 observations while the long series have more than 120 observations. The contenders of the NN3 competition had to forecast the next 18 future months for all the series.

In this section the NN3 competition data has been used to assess the performance of the strategies for preprocessing and modeling explained in Sects. 4 and 5. The goal is to select those strategies that are efficient and effective to develop an automatic methodology for time series forecasting using k-NN regression. Because we have described so many strategies, it is very difficult to assess every possible combination. Instead, we will compare the different strategies for a common task, e.g. for trend preprocessing, and will select one, so that our methodology is built incrementally. For example, we will compare the strategies for selecting k and, once one approach has been selected, we will decide how to select the input variables. In this way, we can develop a suitable methodology—although maybe not the best—and detect the role that each strategy plays in it.

In order to evaluate a strategy we will compute its associated SMAPE. The SMAPE is a measure for assessing forecast accuracy that calculates the symmetric absolute error in percent. Given the forecast F for a NN3 time series with actual values X:

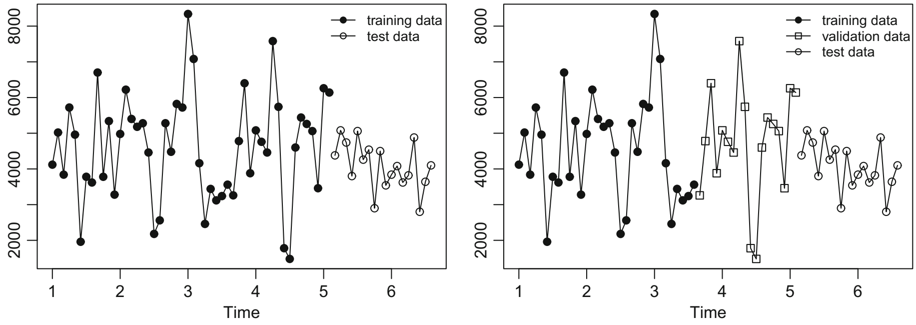

Fig. 7 Using a validation set

$$SMAPE = \frac{1}{18}\sum_{t=1}^{18}\frac{|X_t - F_t|}{(|X_t| + |F_t|)/2}100$$

The SMAPE of each series will then be averaged over all the 111 time series for a global mean SMAPE. Although some experts discourage the use of SMAPE (Hyndman and Koehler 2006), it was the main measure for assessing forecast accuracy in the NN3 competition, so we decided to use it to compare our results with the NN3 contenders.

We have also done pairwise comparisons between strategies using the Wilcoxon signed-rank test (Wilcoxon 1945) to find statistically significant differences.

Some of the strategies analyzed in this section use a validation set to decide whether the strategy is used or to choose some parameters. In this case, the last 18 observations of the training set are used as validation set—see Fig. 7.

In the following subsections the preprocessing and modeling strategies of our methodology are selected. We start with the modeling strategies.

6.1 Selecting the modeling strategies

In this section the performance of the different modeling strategies explained in Sect. 5 is assessed.

6.1.1 Selecting k

In order to select k the following strategies are compared—see Sect. 5.1:

1. k is the square root of the number of training instances.
2. k is chosen by optimizing the SMAPE on a validation set of 18 observations using rolling origin evaluation.
3. Three models, with k equals to 3, 5 and 7 respectively, are used and their forecasts averaged. The values 3, 5 and 7 have been selected after an exhaustive experimentation. The experimentation consists in trying every possible combination of three k-NN models with k ranging from 2 to 14.

To compare these strategies we have used a k-NN regression model with no preprocessing, using lags 1–12 as input variables and the recursive approach as multi-step ahead approach. The results are included in Table 1. In this table the third column shows the global mean SMAPE achieved by each strategy over the 111 time series of the NN3 competition. The

Table 1 Comparison of strategies for selecting k

#	Approach	SMAPE	Time	Comparison	p value
1	Square root	17.3	0.38	1 versus 2	0.35
2	Optimization	17.15	47	1 versus 3	0.17
3	Combination	16.94	0.62	2 versus 3	0.58

Table 2 Comparison of distance metrics

#	Distance metric	SMAPE	Comparison	p value
1	Euclidean distance	16.94	1 versus 2	0.4
2	Manhattan distance	17.13		

fourth column is the time in seconds needed to forecast all the series. The last two columns show pairwise comparisons of the strategies using the Wilcoxon signed-rank test. As can be observed from the p values, there is no strategy statistically significantly better than the others.

In this situation, we have selected the combination approach for our methodology. Apart from obtaining the best results, we think that a combination can be more robust than a model based on an unique k. It is worth noting that the optimization strategy is quite slow in comparison to the other strategies. Furthermore, if the value of k depends on its behavior on a validation set, its selection should be integrated in the selection of the input variables, slowing even more the time consuming process of input selection.

6.1.2 Selecting the distance metric

As commented in Sect. 5.4, to our knowledge all the works using k-NN for time series forecasting utilize the Euclidean distance as the metric to find the nearest neighbors. Anyway, in this section a second distance metric, the well-known Manhattan distance—where the absolute differences between attribute values are added up—, is compared with the Euclidean distance. To compare these distance metrics we have used a k-NN regression scheme based on no preprocessing and the use of three models with different values of k—3, 5 and 7—whose forecasts are averaged to generate the final forecast. As input variables we have used the first 12 consecutive available variables. Table 2 shows the result of the comparison. No distance metric is statistically significantly better than the other. We have chosen the Euclidean distance because is the common distance metric used in time series forecasting with k-NN regression.

6.1.3 Selecting how to combine the k targets

In this section the strategies described in Sect. 5.4 for combining the k targets of the nearest neighbors are compared. The comparison has been done using a k-NN regression scheme based on no preprocessing and the use of three models with different values of k—3, 5 and 7—whose forecasts are averaged to generate the final forecast. As input variables we have used the first 12 consecutive available variables. Table 3 shows the result of the comparison. The combination using a weighted average is statistically significantly worse than the others. The other ways of combining the targets obtain similar results, so the mean has been selected because it obtains the best accuracy.

Table 3 Comparison of strategies for combining the k targets

#	Approach	SMAPE	Comparison	p value	Comparison	p value
1	Mean	16.94	1 versus 2	0.23	2 versus 4	0.39
2	Median	17.05	1 versus 3	< 0.01	3 versus 4	< 0.01
3	Weighted mean	21.5	1 versus 4	0.47		
4	Trimmed mean	17.03	2 versus 3	< 0.01		

Table 4 Comparison of multi-step ahead strategies

#	Approach	SMAPE	Time	Comparison	p value
1	Direct	17.35	1	1 versus 2	0.21
2	Recursive	16.94	0.62	1 versus 3	4e–05
3	MIMO	18.34	0.31	2 versus 3	0.005

6.1.4 Selecting the multi-step ahead forecasting strategy

In the NN3 competition 18 future points have to be forecast. In this section we compare the performance of three approaches for implementing the multi-step ahead forecasting strategy: (1) direct, (2) recursive and (3) MIMO—see Sect. 5.3.

To compare these approaches we have used a k-NN regression scheme based on no preprocessing and the use of three models with different values of k—3, 5 and 7—whose forecasts are averaged to generate the final forecast. As input variables we have used the first 12 consecutive available variables. Table 4 shows the result of the comparison. The MIMO strategy is statistically significantly worse than the other two. We have therefore to select between the direct and recursive strategy without any statistical evidence. We have chosen the second one because is simpler and slightly faster.

6.1.5 Selecting the input variables

In this section the following approaches for selecting the input variables are compared—see Sect. 5.2:

1. Lags 1–12. We have used twelve consecutive lags because the frequency of the series in the NN3 competition is monthly.
2. Lags with significant autocorrelation in the PACF. If a time series has no lag with significant autocorrelation, lags 1–12 are used.
3. Forward selection
4. Backward elimination
5. Forward-backward selection

To compare these strategies we have used no preprocessing and the recursive approach for multi-step ahead forecasting. We have also used three models with different values of k—3, 5 and 7. In the wrapper approaches the maximum lag considered as input variable is 12. The result of the comparison is presented in Table 5. No strategy is statistically significantly better than the others. The wrapper approaches are more time consuming. However, they achieve better results and are recommended in other studies (Sorjamaa et al. 2007), so we select one of them—forward selection—as our method for choosing the input variables.

Table 5 Comparison of strategies for selecting explanatory variables

#	Approach	SMAPE	Time	Comparison	p value	Comparison	p value
1	1:12	16.94	0.62	1 versus 2	0.57	2 versus 4	0.60
2	PACF	16.86	0.5	1 versus 3	0.32	2 versus 5	0.96
3	FS	16.19	89	1 versus 4	0.88	3 versus 4	0.63
4	BE	16.35	120	1 versus 5	0.31	3 versus 5	0.41
5	SR	16.2	91	2 versus 3	0.99	4 versus 5	0.61

Table 6 Comparison of including or not outlier treatment

Approach	SMAPE
No treatment	16.19
Treatment	16.14

6.2 Selecting the preprocessing approaches

In this section the performance of the preprocessing strategies explained in Sect. 4 is evaluated to decide whether they should be included into our k-NN regression methodology. As it was stated in Sect. 4.1 the time series will not be normalized.

Most of the strategies for preprocessing are applied conditionally based on their results on a validation set. If a strategy is effective on the validation set, then it is applied on the test set; if not, its use is discarded for the test set. As a side effect of applying techniques conditionally, the Wilcoxon signed rank test has not been used. The reason is that if a model is compared with itself with a preprocessing applied conditionally, when the condition is not met both models are identical and therefore generate the same results. In this situation, the comparison generates a lot of ties and the Wilcoxon signed rank test is unable to produce a p value.

6.2.1 Outlier treatment

In this section we have evaluated the use of the strategy explained in Sect. 4.2 to detect an remove outliers. The results are presented in Table 6. At first sight, the effects of detecting and removing outliers seem to be negligible. However, it is worth noting that outliers are rare; the strategy detects outliers in only 4 out of the 111 time series of the NN3 competition. Therefore, it seems that this strategy is effective and it will be included in our methodology.

6.2.2 Box–Cox transformation

In this section we assess the use of a Box–Cox transformation to preprocess the time series—see Sect. 4.3. Of course, the forecasts on the transformed series have to be back-transformed. To choose the value of the λ parameter we have used the function BoxCox.lambda from the R forecast package (Hyndman and Khandakar 2008). We have compared the following alternatives:

1. Not using a Box–Cox transformation.
2. Using a Box–Cox transformation unconditionally.

Table 7 Comparison of Box–Cox transformation

Approach	SMAPE
No transformation	16.15
Unconditionally transform	16.50
Conditionally transform	16.13

Table 8 Comparison of strategies for detendring

Approach	SMAPE
No strategy	16.15
Differencing	17.86
STL decomposition	17.15

3. Using a Box–Cox transformation conditionally based on getting better results on the validation set than if no Box–Cox transformation is used.

Table 7 shows the results of the comparison. The best results are achieved by the conditional transformation. We have decided, however, not to use a Box–Cox transformation in our methodology for the following reasons: (1) the difference between not transform and transform conditionally is negligible, (2) not to transform is faster, (3) after a visual inspection of the NN3 competition time series multiplicative seasonality is not an issue.

6.2.3 Trend

In this section we assess the effectiveness of the strategies explained in Sect. 4.4 to detrending a time series. We have analyzed the following alternatives:

1. Not using a detrending strategy.
2. Using differences conditionally.
3. Using a STL decomposition conditionally.

As in the case of the Box–Cox transformation, when we specify that a preprocessing technique is applied conditionally we mean that a validation set is used to test whether the preprocessing technique is effective. The result of the comparison is shown in Table 8. The strategies for dealing with a trend do not seem to be effective. In order to find the reason behind these poor results we have further analyzed the trend pattern in the series of the NN3 competition. We have found that only a 6% of the test points are out of the range of historical observations. Almost the half of these points are in the series 77, 81 and 82. These series are shown in Figs. 8, 9 and 10. Series 77 presents a trend pattern in the historical data that goes on, though damped, in the test data. However, series 81 and 82 are very difficult to predict by any detrending scheme because their historical observations do not seem to have any trend; series 81 presents a level shift in the test data and series 82 an strong upward trend. From this study of the trend pattern, we can conclude that the NN3 competition data is not a good choice for assessing a detrending strategy. Therefore, we will not include any detrending scheme in our methodology for forecasting NN3 series.

6.2.4 Seasonality

In this section the strategies analyzed in Sect. 4.5 for addressing seasonality are evaluated. We have compared the following approaches:

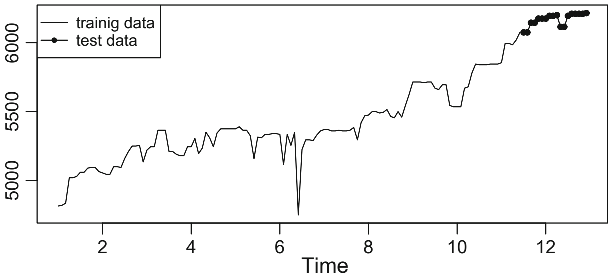

Fig. 8 Time series 77 from NN3 competition

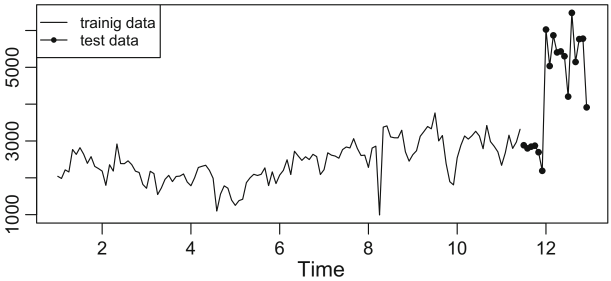

Fig. 9 Time series 81 from NN3 competition

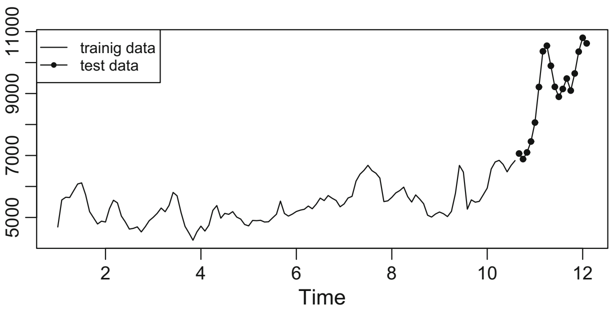

Fig. 10 Time series 82 from NN3 competition

1. Not using a deseasonalizing strategy.
2. Using differences conditionally.
3. Using the strategy devised by Yan (Yan 2012).

Results are presented in Table 9. Again, the best strategy consists in doing no preprocessing—in spite of the fact that more than half of the series in the NN3 competition have a seasonal pattern. Figure 11 shows time series 56 from the NN3 competition and its forecast using the model without preprocessing. The model capture the seasonal pattern very well. Therefore, as our experimentation suggests that *k*-NN is able to deal with seasonal patterns without preprocessing we have decided not to include any strategy for dealing with seasonality in our methodology.

Table 9 Comparison of strategies for dealing with seasonality

Approach	SMAPE
No strategy	16.15
Differencing	17.31
Yan	16.33

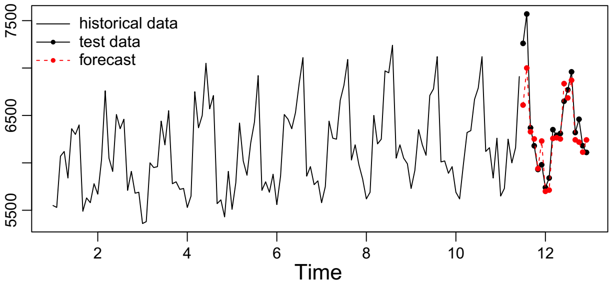

Fig. 11 Time series 56 and its forecast

```
1. Detect and remove outliers
2. Use forward selection on a validation set to select input variables I
3. Use I to build three k-NN models with k = 3, 5, 7
4. Every model generates its forecasts using the recursive approach
5. The final forecast is the mean of the forecasts of the three models
```

Fig. 12 Methodology

7 Methodology

After experimenting in Sect. 6 with different preprocessing and modeling schemes, we have come to the following methodology for applying k-NN, see Fig. 12. Taking into account the predominant role that the selection of input variables plays in forecast accuracy (Zhang et al. 1998) and the good results obtained by the combination of methods, we have devised an alternative methodology in which we use a combination of models, like the one in Fig. 12, where every model uses a different scheme for selecting the input variables. Three models will be combined and each model will use a different wrapper approach—forward selection, backward elimination and forward-backward selection—for selecting the input variables, since the wrapper approaches achieved the best results.

Table 10 presents a performance comparison of our methodology with the top-performed models evaluated in the NN3 competition—in that competition 63 models were evaluated. In the first column of the table, C stands for computational intelligence model and B stands for statistical benchmark. The SMAPE is the average SMAPE on the 111 time series. Both models, the presented in Fig. 12 and the combination described previously in this section, are only outperformed by one computational intelligence method. The combination model reaches the fifth overall position and both proposed models outperform the k-NN model submitted by D'yakonov to the original competition.

Because the NN3 competition used four categories of time series to assess the performance of the methods on different data conditions, we have evaluated our combination methodology on the different data sets: short seasonal, short nonseasonal, long seasonal and long

Table 10 Our methods compared with top NN3 contenders

ID	Method	SMAPE
B09	Wildi	14.84
B07	Theta	14.89
C27	Echo state networks	15.18
B03	ForecastPro	15.44
–	Combination	15.78
B16	DES	15.90
B17	Comb S-H-D	15.93
B05	Autobox	15.95
–	Our model	16.15
C03	Linear model + GA	16.31
B14	SES	16.42
B15	HES	16.49
C46	Regression tree ensemble	16.55
C13	k-NN (D'yakonov)	16.57

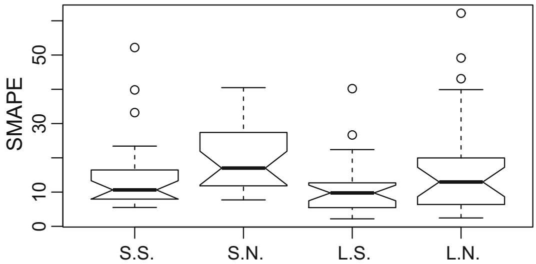

Fig. 13 Boxplots of SMAPEs grouped by time series category

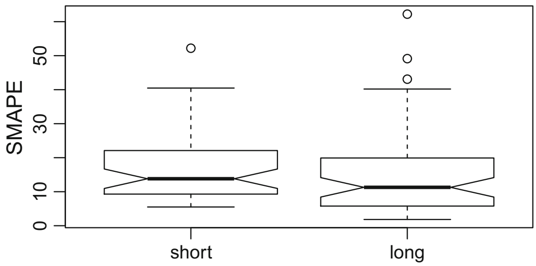

Fig. 14 Boxplots of SMAPEs according to time series length

nonseasonal—see Fig. 13. It seems that our methodology works better with the seasonal series. As for the length of the time series, the boxplots in Fig. 14 shows a comparison. Long time series get better results. Finally, the boxplots in Fig. 15 show the forecast accuracy for different forecast horizons, as the forecast horizon increases the accuracy decreases, which is the expected behavior.

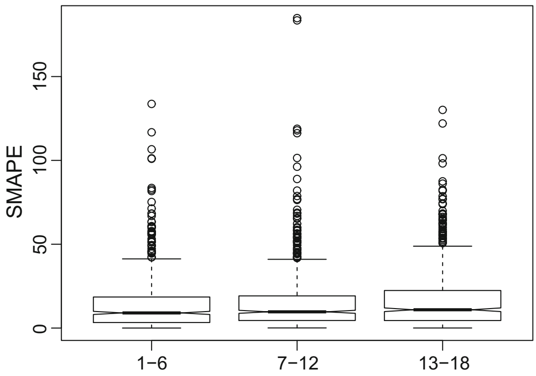

Fig. 15 Boxplots of SMAPEs according to forecast horizon

8 Summary

In this paper we have tried to develop an efficient, effective and automatic methodology for applying k-NN regression on a time series forecasting setting. To reach this goal, we have experimented with the effectiveness of several modeling and preprocessing techniques using the NN3 competition data set.

Although we have not found almost any statistically significant difference between techniques, the techniques selected have proven to be effective in forecasting the NN3 competition data set. Furthermore, they are easy to implement and computationally efficient, which added to the simplicity of k-NN produces an overall straightforward methodology.

As a result of the experimentation, modeling is clearly more important than preprocessing in our methodology. Specifically, the selection of input variables is a key factor in the forecast accuracy of our proposal. Deseasonalizing techniques seem to be superfluous since k-NN appears to be able to deal with seasonal patterns. On the other hand, detendring techniques are necessary because k-NN cannot model a global trend. Unfortunately, we have found that the NN3 competition data set is not suited to assess detendring approaches. Anyway, the analyzed techniques do not seem to be very promising.

One interesting feature of our methodology is the way k is selected. Instead of using an optimization tool, a combination of models with preset k parameters has been used. This solution is both efficient and effective. Similarly, we have also proposed an alternative methodology that uses a combination of models with different schemes for selecting input variables. The alternative methodology seems to improve forecast accuracy. In general, when several design alternatives are attractive, it is worth trying to combine them.

As future work, we would like to study other detendring schemes and use a data set more suitable for assessing time series with global trends. Also, we would like to try to develop forecasting methodologies for artificial neural networks.

References

Ahmed NK, Atiya AF, Gayar NE, El-shishiny H (2010) An empirical comparison of machine learning models for time series forecasting. J Econ Rev 29(5–6):594–621

Al-Qahtani FH, Crone SF (2013) Multivariate k-nearest neighbour regression for time series data—a novel algorithm for forecasting UK electricity demand. In: IJCNN

Bates JM, Granger CWJ (1969) The combination of forecasts. Oper Res Q 20:451–468

Ben Taieb S, Bontempi G, Atiya AF, Sorjamaa A (2012) A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert Syst Appl 39(8):7067–7083

Box GEP, Jenkins GM, Reinsel GC (2008) Time series analysis: forecasting and control, 4th edn. Wiley, Hoboken

Cleveland RB, Cleveland WS, McRae JE, Terpenning I (1990) STL: a seasonal-trend decomposition procedure based on loess. J Off Stat 6(1):3–73

Crone SF, Hibon M, Nikolopoulos K (2011) Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. Int J Forecast 27(3):635–660

Fernandez-Rodriguez F, Sosvilla-Rivero S, Andrada-Felix J (1999) Exchange-rate forecasts with simultaneous nearest-neighbour methods: evidence from the EMS. Int J Forecast 15(4):383–392

Freitas AA (2002) Data mining and knowledge discovery with evolutionary algorithms. Springer, New York

Hibon M, Evgeniou T (2005) To combine or not to combine: selecting among forecasts and their combinations. Int J Forecast 21(1):15–24

Hyndman R, Athanasopoulos G (2014) Forecasting: principles and practice. OTexts

Hyndman R, Khandakar Y (2008) Automatic time series forecasting: the forecast package for R. J Stat Softw 27(1):1–22

Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. Int J Forecast 22:679–688

Hyndman RJ, Koehler AB, Ord JK, Snyder RD (2008) Forecasting with exponential smoothing: the state space approach. Springer, Berlin

Lora AT, Santos JCR, Ramos JLM, Santos JR, Expsito AG (2003) Influence of *k*NN-based load forecasting errors on optimal energy production. In: Moura-Pires F, Abreu S (eds) EPIA, lecture notes in computer science, vol 2902. Springer, Berlin, pp 189–203

Makridakis S, Hibon M (2000) The M3-competition: results, conclusions and implications. Int J Forecast 16(4):451–476

Ord K, Fildes R (2003) Principles of business forecasting. South-Western, Nashville

Ren Y, Suganthan P (2014) Empirical mode decomposition-*k* nearest neighbor models for wind speed forecasting. J Power Energy Eng 2:176–185

Sorjamaa A, Hao J, Reyhani N, Ji Y, Lendasse A (2007) Methodology for long-term prediction of time series. Neurocomputing 70(16–18):2861–2869

Tashman LJ (2000) Out-of-sample tests of forecasting accuracy: an analysis and review. Int J Forecast 16(4):437–450

Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics 1(6):80–83

Witten IH, Frank E, Hall MA (2011) Data mining: practical machine learning tools and techniques, 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco

Yakowitz S (1987) Nearest-neighbour methods for time series analysis. J Time Ser Anal 8:235–247

Yan W (2012) Toward automatic time-series forecasting using neural networks. IEEE Trans Neural Netw Learning Syst 23(7):1028–1039

Zhang G, Eddy Patuwo B, Hu YM (1998) Forecasting with artificial neural networks: the state of the art. Int J Forecast 14(1):35–62

Zhang N, Lin A, Shang P (2017) Multidimensional *k*-nearest neighbor model based on EEMD for financial time series forecasting. Physica A 477:161–173