Forecasting for big data

Nikolopoulos, Konstantinos; Petropoulos, F.

Computers and Operations Research

DOI:
10.1016/j.cor.2017.05.007

Published: 01/10/2018

Peer reviewed version

Forecasting for big data: does suboptimality matter?

Konstantinos Nikolopoulos[a,*], Fotios Petropoulos[b]

[a]*Bangor Business School, Bangor University, UK*
[b]*School of Management, University of Bath, UK*

Abstract

Traditionally, forecasters focus on the development algorithms to identify optimal models and sets of parameters, optimal in the sense of within-sample fitting. However, this quest strongly assumes that optimally set parameters will also give the best extrapolations. The problem becomes even more pertinent when we consider the vast volumes of data to be forecast in the big data era. In this paper, we argue if this obsession to optimality always bares the respective fruits or do we spend too much time and effort in the pursuit of it. Could we better off by targeting for faster and robust systems that would aim for suboptimal forecasting solutions which, in turn, would not jeopardise the efficiency of the systems under use? This study throws light to that end by means of an empirical investigation. We show the trade-off between optimal versus suboptimal solutions in terms of forecasting performance versus computational cost. Finally, we discuss the implications of suboptimality and attempt to quantify the monetary savings as a result of suboptimal solutions.

Keywords: forecasting, big data, optimisation, retail

1. Introduction

Every forecasting story starts with the same ritual: an excerpt from the renowned M-Competitions (Makridakis et al., 1982, 1993; Makridakis and Hibon, 2000). The forecasting competitions that from the early 80s to late 90s road-mapped the basic principles of forecasting; with the first one being: "statistically sophisticated or complex methods do not necessarily provide more accurate forecasts than simpler ones".

The story remains by and large the same up to today - with even the latest state of the art research supporting the assertion. Ghandara et al. (2016) provided further empirical evidence that nature-inspired optimization routines embedded in complex models do not necessarily lead to any performance improvement, if any. They demonstrate that under the volatility and uncertainty met in most financial markets, complex prediction models are on

*Correspondance: K Nikolopoulos, Bangor Business School, Hen Goleg, angor University, College Road, Bangor, LL57 2DG, UK.

Email addresses: k.nikolopoulos@bangor.ac.uk (Konstantinos Nikolopoulos), f.petropoulos@bath.ac.uk (Fotios Petropoulos)

par or worse than more simple models in out-of-sample forecasting evaluation and they urge for future research to focus on the conditions under which computer intelligence optimization methods are being utilized in practice.

In fact forecasting as a discipline has not moved forward much since these research milestones were achieved back in the 80s and 90s. And that despite the call for action from the very originator of the field, Professor Spyros Makridakis. In an interview for the International Journal of Forecasting (Fildes and Nikolopoulos, 2006), he urged for seizing the power provided from super-intelligent and super-fast ICT systems in order to see, analyse and forecast data in a much better way. In a way, he opened the "forecasting for big data" agenda much earlier and asked the pure ICT potential to be harnessed for better forecasting capabilities in practice. The theory, however, was there anyway for many decades in the form of advanced data mining and knowledge extraction algorithms (Haykin, 1998; Härdle, 1992; Haykin, 2008; Heaton, 2012).

This lack of progress is not attributed to neither the lack of IT/ICT power nor the (non-) advance of respective algorithms: it is all down to 21st-century business environment being so volatile that only robust and fast adaptive methods can provide good forecasts over a long period of time. This last point is very important as we need accurate forecasts for each and every decision (and thus) forecasting period. So, one-off "wonder" forecasting methods are not good in real life; robustness is a key element.

Another key point for methods to be successful is to be simple, as per the opening quote, but also being adaptive and able to be tuned fast for the respective performance. Methods that over learn and overoptimise training data sets are not good enough in real life contexts (Haykin, 1998), as the in-sample learning follows a U-shape function so that after a point over-training leads to negative in-sample and even worse out-of-sample performance.

This is exactly what our contribution from this research is aspiring to corroborate. What is the extent of optimality that we should aim for when training and selecting respective parameters in forecasting methods? We want to explore in-sample optimised suboptimal parameter selection of forecasting models, and sequentially quantify the impact, if any, on out-of-sample forecasting accuracy metrics.

We consider as an illustrative example the context of retails operation management: retailers handle from a few hundred products (in a local store), to a few thousands in a local Tesco Express store, to 100,000 SKUs (Stock Keeping Units) in a large Sainsbury's store in UK. The replenishment frequency can be from several hours for fast moving consumer goods like milk and vegetables, to weekly for stationery etc. The hierarchy dictates that orders are set at local shop level but supply is decided at the distribution centre level and the method to provide forecasts range from very basic extrapolations methods like Naive and Moving Averages (Makridakis and Hibon, 2000), to very computational intensive methods with ANNS, Genetic Algorithms and swarm intelligence (Haykin, 1998). This is an example of "big data" in terms of more the sheer volume of information that has to be handled, rather than the "richness" of it - in terms of exploratory variables that can drive demand. In such contexts any savings that can be achieved is important, and to that end suboptimal parameter selection could save a lot of computational time in forecasting support systems, as will be evidenced in our study.

In this research study, we explore suboptimality by considering a simple forecasting method (Simple Exponential Smoothing) and two optimisation approaches (grid-search and trial and error). Using a subset of the M3-competition data, we demonstrate the effects of suboptimality on forecast accuracy, namely the symmetric Mean Absolute Percentage Error and, consequently, the statistical differences in the performance rankings. We trade-off forecast accuracy against the computational time required for producing optimal versus suboptimal models. The next section discusses the data used in this study and the experimental design that was implemented. Section 3 presents the numerical results. Section 4 provides a short discussion of the results as well as implications for theory, practice and software vendors. Section 5 concludes the study.

2. Design

In order to explore the effects of optimality and suboptimality on the forecasting performance, we use the monthly industry subset from the M3-competition (Makridakis and Hibon, 2000). The M3-competition is the largest up-to-date forecasting competition, featuring a total of 3,003 time series of various categories (micro, macro, demography, finance, industry and other) and frequencies (yearly, quarterly, monthly and other). In the original study (Makridakis and Hibon, 2000), 24 methods and commercial packages were compared with regards to their forecasting performance. Since then, the data has been used numerous times for research purposes, and the development of new forecasting methods. The industry monthly subset consists of 334 time series of varying lengths, however in all cases the available history spreads for at least eight years with a mean value of twelve years. The exact lengths and respective number of time series are presented in table 1. In their majority, the data represent either sales or demands and, as such, can be considered a good proxy for retail data.

Table 1: Length of the available monthly industry time series.

Number of observations	96	122	128	133	134	136	137	139	140	141	142	143	144
Number of time series	1	1	1	58	46	2	1	9	6	12	5	7	185

The forecasting function that is implemented in this study is the simplest form of the exponential smoothing family, the simple exponential smoothing (SES) method. SES is very widely used in practice and is suitable for data that do not exhibit trend or seasonality. It is based on an exponential smoothing average, where more recent observations are assigned larger weights. The degree of the smoothness is controlled via a smoothing parameter, α, which takes values in the range $[0, 1]$. The one-step-ahead forecast of the exponential smoothing method is calculated as $f_{t+1} = \alpha y_t + (1 - \alpha) f_t$, where y_t represents the actual value at period t and f_t the forecast for the respective period. If forecasts for further horizons are required, these are equal to the one-step-ahead forecast, $f_{t+h} = f_{t+1}$, as SES produces flat forecasts.

In this work we study the effects of optimising (or suboptimising) the α smoothing parameter. The initial forecast (also called initial level) is not optimised, rather it is set

equal to the initial actual observation, or $f_1 = y_1$. The algorithmic implementation (in R language) of SES that we used in this study is provided in Appendix A.

Two simple optimisation methods are considered. The first one is widely known as grid-search optimisation (also known as parameter sweep or exhaustive search). Keeping in mind that the parameter to be optimised can take values within a certain range (in our case α takes values in $[0, 1]$), the algorithm starts from the one end of the range and reaches the other end after n steps. Essentially, all possible values of the parameter within the range are considered with an updating interval that equals to $m = |max - min|/n$, where max and min correspond to the limits of the range. For example, if $n = 100$, then every α value with two decimal points is tested $(0, 0.01, 0.02, ..., 1)$. For each value of the smoothing parameter, the corresponding one-step-ahead forecasts are calculated and the model fit is measured by the means of the Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_i)^2 \tag{1}$$

Other error measures could be considered (such as the Mean Absolute Error, or MAE), however the MSE is the most widely used in practice. Effectively, $n+1$ MSEs are calculated and the smoothing parameter with the lowest MSE is considered to be the optimal one. The algorithmic implementation of the grid-search optimisation is provided in Appendix A.

The second optimisation algorithm, that we consider in this study, is the trial and error algorithm, which is a fixed-step convergence procedure through a modified Luus-Jakola approach. The search of the optimal α smoothing parameter starts from the values 1/3 and 2/3 where the corresponding MSEs are calculated. The smoothing value with the lowest MSE is selected as the current optimal ($\dot{\alpha}$). For every subsequent step ($k = 2, 3, ...$), the algorithm calculates the MSE that corresponds to the smoothing values with distance $\frac{1}{3\times 2^{k-1}}$ from the current optimal, or $\dot{\alpha} \pm \frac{1}{3\times 2^{k-1}}$. Among the smoothing values $\dot{\alpha} - \frac{1}{3\times 2^{k-1}}$, $\dot{\alpha}$ and $\dot{\alpha} + \frac{1}{3\times 2^{k-1}}$, the one with the lowest MSE is selected as the new current optimal. This procedure is repeated for a pre-specified number of steps n, with $k \leq n$. Effectively, $2n$ MSEs are calculated, while the trial and error approach is expected to work well when MSE is a U-shape function of the α smoothing parameter. The algorithmic implementation of the trial and error optimisation is provided in Appendix A.

For each of the two optimisation methods discussed above, ten cases are considered with regards to the value of n. Table 2 provides the number of steps (n) considered in each case. It is worth noting that the number of steps has a direct impact on the subsequent accuracy in identifying the true optimal smoothing parameter that minimises the one-step-ahead in-sample forecast error. In other words, we can simply assume that by terminating the process of searching for an optimal α smoothing parameter in less steps, then a suboptimal value is selected.

As the majority of the available time series exhibit seasonality, which cannot be modelled by SES, we consider a seasonal adjustment of the data. We follow the procedure applied in the Theta method by Assimakopoulos and Nikolopoulos (2000) and described in detail by

Table 2: Number of steps (n) considered for each optimisation method and each case.

Case	1	2	3	4	5	6	7	8	9	10
Grid-Search	1	2	3	5	10	20	100	200	1000	10000
Trial and Error	1	2	3	4	5	6	7	8	9	10

Fioruci et al. (In press). First, each time series is tested for significant seasonal behaviour by the means of the autocorrelation function for lag equal to the number of periods per year (twelve for monthly data). We opt for 90% confidence level. If the series is identified as seasonal, then a multiplicative classical decomposition is applied on the data and the seasonal component is removed. Both forecasting and evaluation are performed on the seasonally adjusted data.

Rolling origin evaluation is performed to produce forecasts and measure the performance of SES under optimal and suboptimal smoothing parameters. The last two years of data (24 observations) of each time series are withhold. We first produce an one-step-ahead forecast from origin $N - 24$, where N represents the length of the time series. Then, one period is added into the in-sample and one-step-ahead forecast is produced from origin $N - 23$. This procedure is repeated 24 times, with the origin $N - 1$ being the last period from where forecasts are produced. This exercise provides us with 24 one-step-ahead forecasts for each time series. However, and given that SES generates flat forecasts, the rolling origin scheme described above allows us to evaluate SES for horizons greater to one period ahead. To do so, we can suitably shift the one-step-ahead forecasts to end up with 23 two-steps-ahead forecasts, 22 three-steps-ahead forecasts and 19 six-steps-ahead forecasts.

3. Results

In this section we evaluate the performance of the two optimisation approaches described in section 2 for the various cases considered. The cases correspond to the number of steps utilised, before an algorithm finishes, which in turn refers to the selection of optimal or suboptimal parameter values. We measure the performance in terms of:

1. Computational time (in seconds); we argue that this is a very important dimension given that in many practical situations nowadays forecasts are required for a very large number of items (usually 100,000 stock keeping units in an average supermarket) while sometimes multiple replenishment cycles can occur within a single day. The forecasts where produced using a cloud machine with 8 virtual computer processing units clocked at 2.1 GHz and 8GB of RAM operating under Windows Server 2008 R2 64-bit operating system. The code was implemented for single-core processing. Regardless of the system used, we expect any relative differences in terms of computational time to hold under different set-ups.
2. Symmetric Mean Absolute Percentage Error (sMAPE); this is a scale-independent error metric suitable for measuring the accuracy of forecasts across different time series. sMAPE is widely used in forecasting research and was the main error measure in the empirical evaluations of the M3-Competition (Makridakis and Hibon, 2000).

sMAPE is defined as

$$sMAPE = \frac{200}{k} \sum_{i=1}^{k} \frac{|y_i - f_i|}{|y_i| + |f_i|}, \tag{2}$$

where the average is calculated across horizons and time series.

3. Multiple Comparisons with the Best test (MCB); the performance (as measured by sMAPE) of the different cases is statistically compared via calculating the average ranks of each method, and constructing the corresponding rank intervals. The null hypothesis of MCB is that the performance across the different cases is statistically indifferent. The null hypothesis is rejected when the constructed ranked intervals for a pair of cases do not overlap. For more details on the MCB test, the reader in encouraged to see Koning et al. (2005).

Figures 1 and 2 present the results of the computational time (x-axis), MCB tests (primary y-axis) and sMAPE (secondary y-axis) for the two optimisation methods (grid-search and trial and error) respectively. The x-axis of Figure 1 is presented in logarithmic scale.

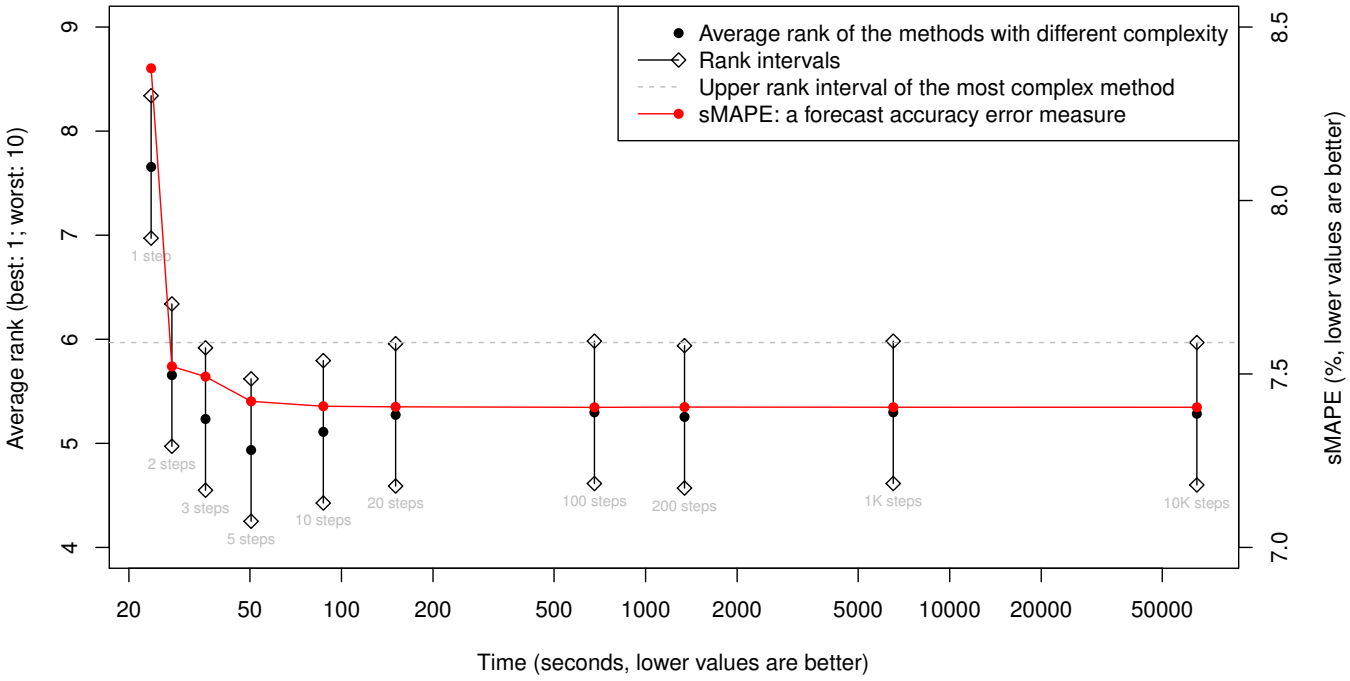

Figure 1: Trade-offs between computational time and forecasting performance for the grid-search method. Note that the x-axis is presented at a logarithmic scale.

Examination of figure 1 (grid-search method) reveals some very interesting insights. In terms of average ranks, only the first case (1 step) is statistically worse than the most time consuming case. In other words, just considering three possible values for α smoothing parameter ($n = 2$, which corresponds to smoothing values 0, 0.5 and 1) results in performance, as evaluated by the average ranks, that is statistically indifferent to searching the optimal value with a step of 0.001 or even 0.0001. Interestingly enough, average ranks for the cases

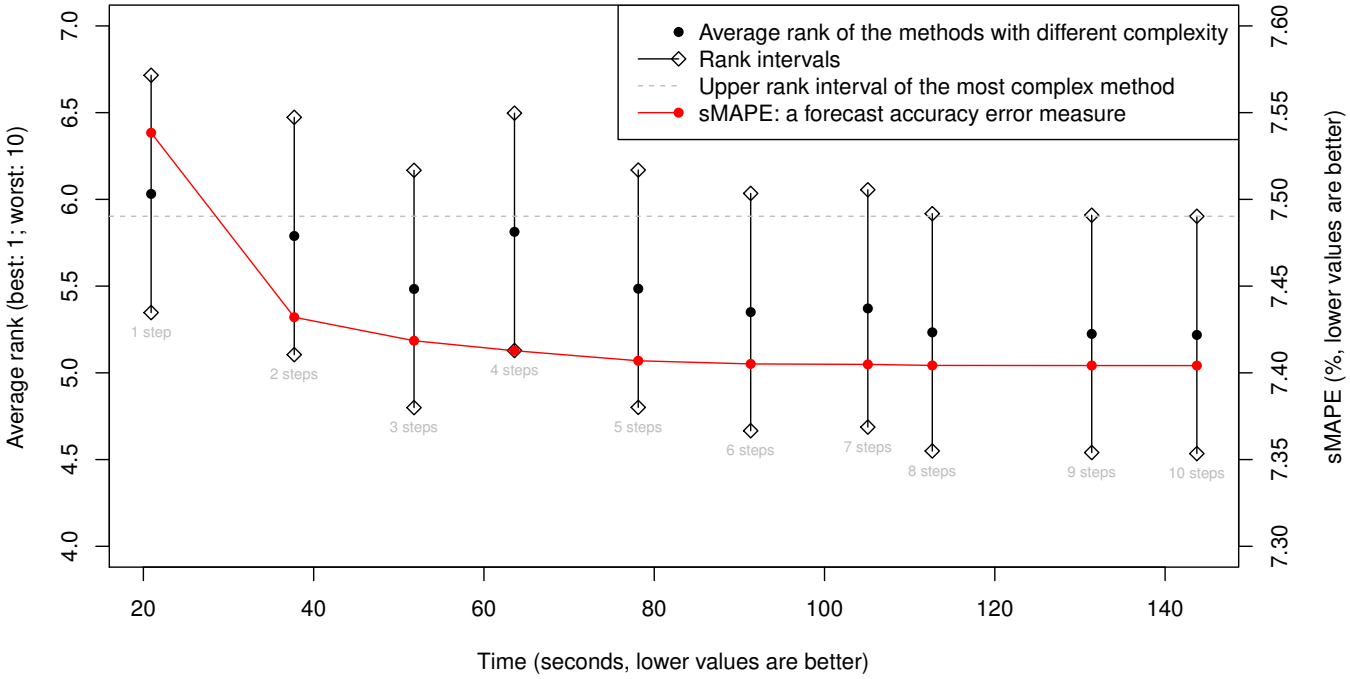

Figure 2: Trade-offs between computational time and forecasting performance for the trial and error method.

where n equals to 5 and 10 are marginally better than more complex and time-intensive cases. In terms of performance as measure by $sMAPE$, the first case scores 8.38%, however the values of $sMAPE$ quickly converge around the level of 7.4% in the cases where $n \geq 5$.

Figure 2 (trial and error method) provides similar insights, however all cases now appear to provide performance that is much closer (from a statistical point of view). All the cases result in not-statistically different average ranks, while differences in terms of sMAPE are smaller. When $n = 1$, $sMAPE = 7.54\%$, however, it quickly converges around 7.4 - 7.45% for $n \geq 2$.

Results from both figures 1 and 2 corroborate to the same conclusion: suboptimally selecting the value of smoothing parameter for SES does not have a negative impact in the one-step-ahead out-of-sample performance. In fact, when $n = 5$ and $n = 3$ for grid-search and trial and error methods respectively, the resulted performance is practically the same as more computational intensive cases. At the same time, suboptimally selecting the smoothing parameter can result in significant gains in terms of computational times. More specifically, the time reductions can be equal to 99.9% and 64% for each of the methods when compared to the most time-consuming cases considered ($n = 10000$ and $n = 10$ respectively). However, significant gains in terms of time (93% and 51%) can also be demonstrated when the widely used industry benchmarks of $n = 100$ and $n = 7$ and considered for grid-search and trial and error respectively.

We also examined the potential effect of the length of the within sample data, where we did not observe any significant differences. However, it is worth-noticing that the lengths of the time series within this data set do not differ significantly, with 99% of the series spanning

between 133 and 144 months.

Another viewpoint of the analysis is provided in figures 3 and 4, where the differences in the selected α parameter value (compared to the most complex case, grid-search method with $n = 10000$) are recorded. We observe that in both optimisation methods the differences are quickly minimised and selected values are practically indifferent when $n = 100$ and $n = 7$ respectively. However, another important insight is revealed. Suboptimally selecting the smoothing parameter via the grid-search method results in biased selections: selected values are significantly larger than the optimal ones. This bias is observed for the cases where $n = 1$ and $n = 2$, however it is not observed when $n \geq 3$. Interestingly, the opposite selection bias is observed for the trial and error method (selected values are lower than the optimal), but this is true only for the simpler case ($n = 1$).

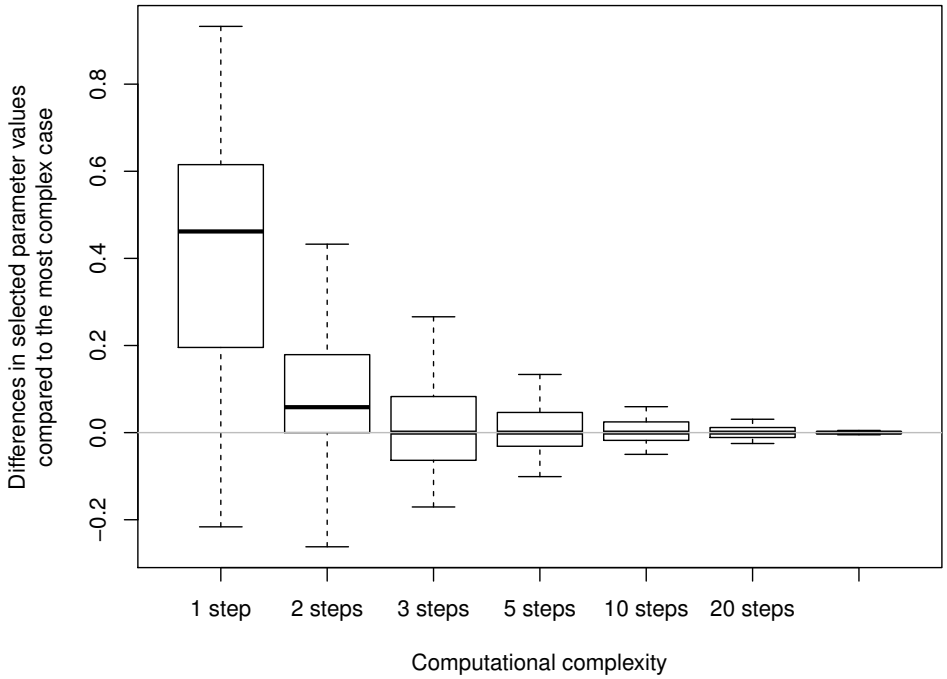

Figure 3: Differences between the selected α smoothing parameter values and the optimal one for the grid-search method.

We also expand the performance results to consider forecasting horizons greater than one. We measure the $sMAPE$ for each optimisation method, each case (1 to 10), and each horizon (1 up to 6 periods ahead). Subsequently, we calculate the percentage differences between the $sMAPE$ of each case compared to the most complex one for the respective optimisation method. For example, the $sMAPE$ for grid-search method where $n = 1$ and horizon equal to 2 is compared to that of $n = 10000$ and the same horizon (2). Finally, we calculate the correlation of these percentage differences with the forecasting horizon. The results for each method and case are presented in table 3.

The results demonstrate strong negative relationships for the simpler of the cases of each optimisation method. This indicates that percentage differences in the values of $sMAPE$

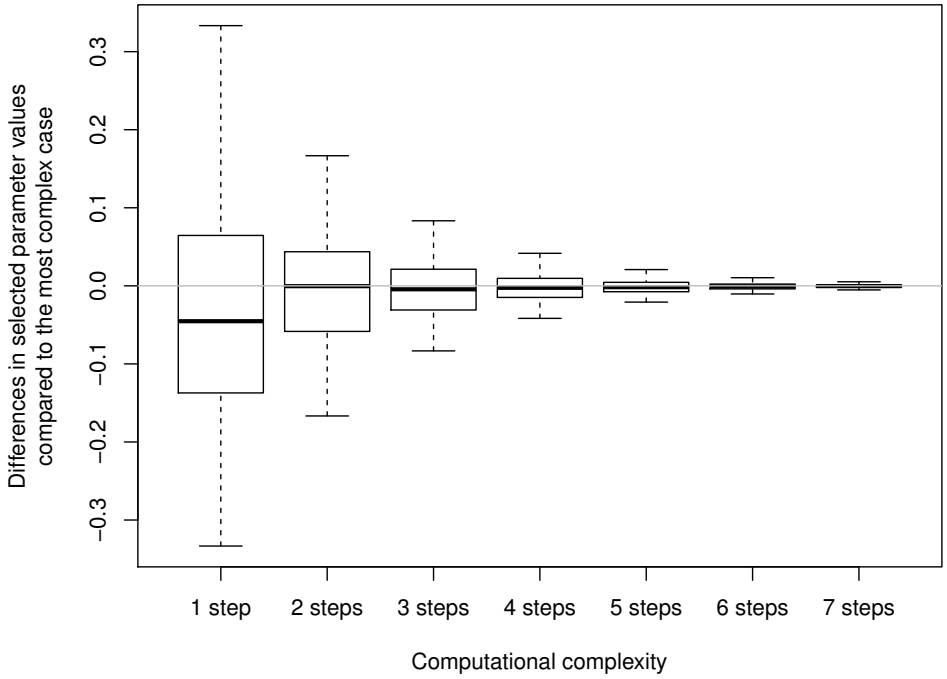

Figure 4: Differences between the selected α smoothing parameter values and the optimal one for the entrainment method.

Table 3: Correlations of percentage differences in sMAPE (compared to the most complex case) with the forecasting horizon.

Case	1	2	3	4	5	6	7	8	9
Grid-Search	-0.808	-0.839	-0.993	-0.958	-0.242	-0.967	-0.345	-0.374	-0.547
Trial and Error	-0.994	-0.960	-0.588	-0.953	-0.990	-0.824	-0.806	-0.396	-0.149

for simpler and the most complex cases decrease as the planning horizon increases. This practically means that suboptimally selecting smoothing parameters has even lesser effects to further horizons.

4. Discussion and implications

Before getting into highlighting the implications from our research, we would like to discuss briefly our thoughts on the empirical results presented in the previous section. For many it might be surprising and counter-intuitive to find that a suboptimal solution actually works that well. The trick here is that in real life the evaluation is done in a different data set than the one the training and the respective optimisation of the models takes place. Assuming that the history repeats itself and that the same patterns pertain the future extrapolation, the room for suboptimal solution becomes very narrow, as one would naturally expect the same set of optimal models (optimised in the past) to keep on producing optimal solutions. But very often in real life the history does not exactly repeats itself, or in some situations not even remotely repeat itself. As such, a primafacie suboptimal (in the past) solution may well perform on-par with optimal (in the past) when the evaluation is done in the future - in a practically new set of observations. This is exactly where we base the explanation of our results: the future is rarely exactly the same as the past, and as such moving around the optimal area (for a set of parameters of the model) does not really harm the observed accuracy of the models in the future, while saving important computational time.

We strongly believe that our investigation presented in this paper has clear implications for theory, practice and software vendors.

- Implications to theory: the quest for optimality is not necessarily the "holy grail" in parameter estimation. In-sample is never exactly the same as what is about to follow on the data front and as such suboptimal solutions may need to be considered when developing theoretical expectation of such parameters. The problem of suboptimality exists even in the simplest forecasting methods, such as the SES as demonstrated in this paper. However we would expect this to be even more evident in a more complex method, such as the Damped exponential smoothing or the Holt-Winter's method where the number of parameters to be optimised (or suboptimised) increases significantly.
- Implications to practice: furthermore for practitioners the choice of suboptimal parameters will result in huge computational time savings without necessarily impacting on forecasting accuracy, so a win-win situation. Expanding on our illustrative example regarding the retails operation management, we assume 100,000 different SKUs for a large store and a machine of similar specifications as the one used for this simulation. The computational times for the proposed suboptimal approaches would be 4.2 hours for grid-search ($n = 5$) and 4.3 hours for trial and error method ($n = 3$) respectively. The respective computational times for the industry benchmarks ($n = 100$ and $n = 7$) would be significantly higher, at 56.5 and 8.7 hours respectively. Given the new trends, many companies rely on cloud computing services, such as the Amazon Web Services

or the Microsoft Azure, which are available to hire by the hour. So, any decrease in computational time can be directly translated to significant monetary savings. Assuming an hourly cost of $0.05 per hour (which is typical for a machine along the specifications used in this research), a retailer that operates 1,000 stores (for comparison, as of September 2016 Tesco operates 6,902 stores), forecasting and replenishment occurring just once a day (even if nowadays shorter replenishment cycles are common practice amongst many retailers), a change from optimality to suboptimality based on the grid-search technique alone could result in annual computational time savings of $950,000.

- Implications to software developers: software designers especially in the Forecasting Support Systems area, should consider offering the option for 'faster suboptimal parameter selection' and let the users decide if they are happy with the forecasting accuracy achieved from such options.

5. Conclusions

Our study is the first, to the best of our knowledge, that explores the impact of suboptimal in-sample parameter selection of forecasting methods on out-of-sample forecasting performance. The evidence provided here makes the case that suboptimal solutions do not produce worse forecasting performance over time, whilst at the same time saving significant computational time - even more important in the era of "big data".

The use of one method and one data set does restrict the full generalisation of our results. But, nevertheless, the fact that the M3-competition is a widely used forecasting benchmark data set and that Bob Brown's simple exponential smoothing is the most used forecasting method in practice, gives a lot of merit in what we presented in this paper. As well as being able to be readily utilised by practitioners in the field, we are confident that extending this study to a wider range of methods and data sets would yield similar results. We suggest that this would be a good avenue for future research.

Finally one more way to advance further the understanding to the agenda we are opening is via trying to investigate the impact on computationally intensive artificial intelligence methods where our intuition suggests that the gains will be even higher. We expect that the same will be true for the longer forecasting horizons where the accuracy levels are expected to be worse overall. All these remain to be empirically evidenced.

Appendix A. Implementations of main functions in R

Simple Exponential Smoothing

```
SES <- function(x, alpha){
  n <- length(x)
  fcs <- array(0, n+1)
  fcs[1] <- x[1]
  for (t in 1:n){
    fcs[t + 1] <- alpha * x[t] + (1-alpha) * fcs[t]
  }
  return(fcs)
}
```

Grid-Search optimisation

```
GridSearch <- function(x, steps){
  bestalpha <- 0
  bestMSE <- 10^10
  for (alpha in seq(0, 1, 1/steps)){
    MSE <- mean((x - SES(x, alpha)[1:length(x)])^2)
    if (MSE < bestMSE){
      bestMSE <- MSE
      bestalpha <- alpha
    }
  }
  return(list(fcs = SES(x, bestalpha)[length(x)+1], alpha = bestalpha))
}
```

Trial and Error optimisation

```
TrialError <- function(x, steps){
  bestalpha <- 0
  bestMSE <- 10^10
  alphas <- c(1/3, 2/3)
  v <- 1/6
  for (step in 1:steps){
    for (i in 1:2){
      MSE <- mean((x - SES(x, alphas[i])[1:length(x)])^2)
      if (MSE < bestMSE){
        bestMSE <- MSE
        bestalpha <- alphas[i]
      }
    }
    alphas <- c(bestalpha + v, bestalpha - v)
    v <- v/2
  }
  return(list(fcs = SES(x, bestalpha)[length(x)+1], alpha = bestalpha))
}
```

References

Assimakopoulos, V., Nikolopoulos, K., 2000. The Theta model: a decomposition approach to forecasting. International Journal of Forecasting 16 (4), 521–530.

Fildes, R., Nikolopoulos, K., 2006. Spyros makridakis: An interview with the international journal of forecasting. International Journal of Forecasting 22 (3), 625–636.

Fioruci, J. A., Pellegrini, T. R., Louzada, F., Petropoulos, F., Koehler, A., In press. Models for optimising the theta method and their relationship to state space models. International Journal of Forecasting.

Ghandara, A., Michalewicz, Z., Zurbruegge, R., 2016. The relationship between model complexity and forecasting performance for computer intelligence optimization in finance. International Journal of Forecasting 32 (3), 598–613.

Härdle, W., 1992. Applied Nonparametric Regression. Cambridge University Press, Cambridge.

Haykin, S., 1998. Neural Networks: A Comprehensive Foundation. Pearson, US.

Haykin, S., 2008. Neural Networks and Learning Machines. Prentice Hall, US.

Heaton, J., 2012. Introduction to the Math of Neural Networks. Heaton Research, Inc.

Koning, A. J., Franses, P. H., Hibon, M., Stekler, H. O., 2005. The M3 competition: Statistical tests of the results. International Journal of Forecasting 21 (3), 397–409.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R., 1982. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. Journal of Forecasting 1 (2), 111–153.

Makridakis, S., Chatfield, C., Hibon, M., Lawrence, M., Mills, T., Ord, K., Simmons, L. F., 1993. The M-2 competition: a real-time judgmentally based forecasting study. International Journal of Forecasting 9, 5–23.

Makridakis, S., Hibon, M., 2000. The M3-competition: results, conclusions and implications. International Journal of Forecasting 16 (4), 451–476.