Operational Research (2022) 22:3037–3061
https://doi.org/10.1007/s12351-020-00605-2

ORIGINAL PAPER

Comparison of statistical and machine learning methods for daily SKU demand forecasting

Evangelos Spiliotis[1] · Spyros Makridakis[2] · Artemios-Anargyros Semenoglou[1,2] · Vassilios Assimakopoulos[1]

Received: 11 June 2020 / Revised: 26 July 2020 / Accepted: 10 September 2020 /
Published online: 16 September 2020

Abstract
Daily SKU demand forecasting is a challenging task as it usually involves predicting irregular series that are characterized by intermittency and erraticness. This is particularly true when forecasting at low cross-sectional levels, such as at a store or warehouse level, or dealing with slow-moving items. Yet, accurate forecasts are necessary for supporting inventory holding and replenishment decisions. This task is typically addressed by utilizing well-established statistical methods, such as the Croston's method and its variants. More recently, Machine Learning (ML) methods have been proposed as an alternative to statistical ones, but their superiority remains under question. This paper sheds some light in that direction by comparing the forecasting performance of various ML methods, trained both in a series-by-series and a cross-learning fashion, to that of statistical methods using a large set of real daily SKU demand data. Our results indicate that some ML methods do provide better forecasts, both in terms of accuracy and bias. Cross-learning across multiple SKUs has also proven to be beneficial for some of the ML methods.

Keywords Forecasting accuracy · SKU demand · Neural networks · Regression trees · Cross-learning

1 Introduction

Daily SKU demand data is typically characterized by irregular demand sizes (erraticness) and variable demand arrivals (intermittency), with many observations having zero values. This is especially true when forecasting at low cross-sectional levels, such as at a store or warehouse level, or dealing with slow-moving items.

✉ Evangelos Spiliotis
spiliotis@fsu.gr

[1] Forecasting and Strategy Unit, School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece

[2] Institute for the Future, University of Nicosia, Nicosia, Cyprus

Thus, effectively forecasting intermittent, lumpy, erratic, and smooth demand series (Syntetos and Boylan 2005) becomes a challenging task, requiring different methods to those used for extrapolating continuous, regular data (Spithourakis et al. 2011; Petropoulos et al. 2013).

However, daily demand forecasting is very common in many industrial and retail settings (Johnston et al. 2003; Willemain et al. 2004; Syntetos and Boylan 2005). Moreover, in many companies, such forecasting must take place for thousands of items and numerous locations (Seaman 2018). Given that inventory management and stock control builds on demand forecasting, with small gains in accuracy leading to considerable inventory reductions (Syntetos et al. 2010) and slight inaccuracies to higher stock holdings and lower service levels than the ones desired (Ghobbar and Friend 2003; Pooya et al. 2019), accurate and computationally affordable demand forecasting becomes critical.

Daily demand forecasting is usually performed by utilizing well-established, statistical forecasting methods, such as the Simple Exponential Smoothing (SES) method (Brown 1959; Gardner 1985), the Croston's method (Croston 1972), and its variants (Syntetos and Boylan 2005; Teunter et al. 2011; Babai et al. 2019). Given the success and the relatively low computational cost of these methods, researchers and practitioners have been working on advancing them further (Boylan and Syntetos 2009; Hasni et al. 2019). However, during the last few years, Machine Learning (ML) methods, and particularly Neural Networks (NNs), have been proposed as an alternative to the statistical ones (Makridakis et al. 2020a). The main advantage of these methods is that they utilize non-linear algorithms capable of learning by trial and error and improving their performance over time by observing the historical data, thus making no or few assumptions about the underlying data generation process (Hornik et al. 1989; Barker 2020).

Nevertheless, in the area of demand forecasting, only a limited number of studies have examined the use of ML methods for forecasting SKU demand time series (Carmo and Rodrigues 2004; Nasiri Pour 2008; Gutierrez et al. 2008; Mukhopadhyay et al. 2012; Kourentzes 2013; Lolli et al. 2017; Nikolopoulos et al. 2016; Boutselis and McNaught 2019), with the majority of them considering only NN methods, a special case of ML. In addition, although most of these studies suggest methodological advances and accuracy improvements, limited objective evidence is available regarding their relative performance as standard forecasting tools. This is mainly due to the lack of sensible, statistical benchmarks and the utilization of small test samples. At the same time, the results of Kaggle's recent forecasting competitions, involving the prediction of daily/weekly sales at product, store, and department level, highlight the potential of ML methods in real-life business forecasting tasks (Bojer and Meldgaard 2020). Thus, comparing the forecasting performance of various ML methods to that of statistical methods becomes vital.

In such a context, Makridakis et al. (2018) recently questioned the superiority of pure ML methods for the case of continuous, regular time series. In their study, the authors evaluated the accuracy of ten popular ML methods and eight statistical ones, concluding that the latter perform better. This conclusion was later supported by the results of the M4 competition (Makridakis et al. 2020b) which, however, also revealed some novel methods that exploit ML algorithms to provide accurate forecasts. What these methods had in common, was that they used information from multiple series to predict individual ones, that way allowing a "cross-learning" modeling approach (Smyl 2020; Montero-Manso et al. 2020).

Taking the above into account, the purpose of this study is to evaluate the performance of popular ML methods for daily SKU demand forecasting and compare their accuracy and bias with that of standard, statistical methods. This would allow the extraction of valuable insights regarding the appropriateness of using ML methods in this special area of forecasting. Since this work is inspired by the study of Makridakis et al. (2018), we consider most of the ML methods used in their study, but add some additional state-of-the-art algorithms, like the ones utilized in relevant forecasting competitions hosted by Kaggle (Bojer and Meldgaard 2020). We choose the statistical benchmarks so that both simple and advanced approaches available in the literature are covered. Our assessment is made using a large dataset that involves daily demand series of 3300 SKUs, sold by a major retail company in Greece.

After evaluating the performance of the examined ML methods, trained in a series-by-series fashion, we proceed by considering a cross-learning modeling approach, as done by the most successful submissions of the M4 competition. The motivation behind this experimentation is that if ML methods are capable of effectively learning from diverse and mostly unrelated data like the one of M4, greater improvements could be achieved when exploiting homogeneous datasets of related SKUs (Makridakis et al. 2020b). Thus, the ML models built earlier, which are trained in a series-by-series fashion and learn from the demand of a single SKU each time, are compared to the corresponding cross-learning ones, learning from the whole dataset concurrently. That way, valuable conclusions can be made about the benefits of considering cross-learning in the area of daily demand forecasting, as well as the types of ML methods that exploit its potential.

The contribution of our study is summarized below:

- We evaluate the performance of ML methods for the case of daily SKU demand forecasting, using a set of sensible statistical benchmarks, both simple and advanced ones.
- In our assessment, we consider a variety of ML methods and not only NNs, as done in the majority of the studies found in the literature.
- We explore the potential of cross-learning in daily SKU demand forecasting by constructing ML models that learn from the whole dataset concurrently and identifying those that perform better than their series-by-series equivalents.
- The improvements reported, both in terms of accuracy and bias, are tested for statistical significance. Moreover, they are summarized for different types of demand time series, indicating the most suitable options for each case. Finally, the trade-off between forecasting performance and computational cost is investigated.

The rest of the paper is organized as follows. Section 2 provides a brief literature review on the ML methods used in demand forecasting and introduces the concept of cross-learning. Section 3 presents the methods utilized in the study, both statistical and ML ones. In Sect. 4, we present the dataset used for evaluating the methods considered and provide the experimental design of the study. The results are presented and discussed in Sect. 5. Lastly, Sect. 6 concludes the study, presents its limitations, and explores avenues for future research.

2 Brief literature review

As noted in the introduction, ML methods have been recently proposed as an alternative to statistical ones for forecasting demand data that may be characterized by intermittency and erraticness. In such settings, more often than not, ML methods are trained in a series-by-series fashion, i.e., a different model is trained per series using its original data as input. For example, this approach has been considered by Carmo and Rodrigues (2004) who compared the performance of Radial Basis Function NNs (RBF) with statistical methods using a sample of 10 irregularly spaced time series, reporting encouraging results. The main advantage of this strategy is that it allows a completely non-linear, assumption-free estimation of forecasts, letting the data speak for themselves. However, in order for this kind of modelling to be effective in practice and capture the dynamics and inter-connections of demand volume and inter-demand intervals, large samples are required. This is because ML methods, in contrast to statistical ones, do not assume an underlying data generation process, making them data hungry. Thus, this approach may be proven inappropriate for demand series that consist of few observations, involve complex patterns, and include lots of zero values (Kourentzes 2013).

In order to deal with this problem, Gutierrez et al. (2008), and later Mukhopadhyay et al. (2012), proposed a simplified feature-based Multi-Layer Perceptron (MLP) model, which uses only the demand at the immediately preceding period(s) and the number of periods separating the last two non-zero demand transactions at the end of the immediately preceding period as inputs. Kourentzes (2013) discussed the limitations of this approach and introduced an interesting alternative which is based on Croston's decomposition approach, but predicts the components of the demand size and inter-demand intervals simultaneously, thus allowing for more flexibility. This approach displayed poor forecasting performance in terms of accuracy and bias, but led to higher service levels than Croston's method and its variants. Nasiri Pour (2008) introduced two other alternatives for implementing NNs in demand forecasting, a feature-based one, and a hybrid. The first approach follows the suggestions of Gutierrez et al. (2008), but expands the features used as input in order to maximize the information considered by the NNs. The second approach mixes a NN that forecasts occurrences of non-zero demands with a recursive method which estimates the quantity of non-zero demands. Both methods reported improvements in terms of accuracy over the considered statistical benchmark. More recently, Lolli et al. (2017) considered NNs trained either by back-propagation or extreme learning machines and compared their performance with that of benchmark neural networks and standard forecasting methods, concluding that back-propagation improves forecasting accuracy, although with increased computational cost. Boutselis and McNaught (2019) used Bayesian Neural Networks (BNN) to forecast spares demand from equipment failures in a changing service logistics context, indicating their superiority over expert adjusted and logistic regression forecasts, while Nikolopoulos et al. (2016) proposed applying nearest neighbor approaches for supply chain data and investigated the conditions under which these perform adequately.

By reviewing the literature we observe that, although ML has attracted the attention of a wide range of practitioners and researchers (Makridakis et al. 2018, 2020a), limited research has been conducted in the field of demand forecasting, focusing mostly on NN methods and approaches for forecasting demand in the presence of promotions (Ali et al. 2009; Abolghasemi et al. 2020), while ignoring other alternatives that could possibly improve forecasting performance, such as Regression Trees (RTs), Support Vector Regression (SVR), and Gaussian processes (GP) (Bojer and Meldgaard 2020). This is exactly where one of the contributions of the present study lies, exploring additional ML methods that have been proven successful in other forecasting applications in the past.

Another important observation is that the majority of the methods utilized in the literature are trained in a series-by-series fashion. As noted, this implementation displays many limitations related to small data samples, slow convergence, and increased computational cost. On the other hand, recent advances in forecasting have shown that models that allow cross-learning, i.e., learning across many series simultaneously, can enhance forecasting performance, without significantly increasing the computational cost (Smyl 2020). For instance, the top two performing methods of the M4 competition (Makridakis et al. 2020b) introduced information from multiple series (aggregated by data frequency) in order to decide on the most effective way of forecasting or selecting the weights for combining the various methods considered. Similarly, the majority of the top performing methods in Kaggle's forecasting competitions, exploited cross-learning by employing advanced ML algorithms, like RTs (Bojer and Meldgaard 2020).

In the context of demand forecasting, Salinas et al. (2020) demonstrated the potential of cross-learning for the case of probabilistic forecasts using autoregressive recurrent networks, Chapados (2014) identified a hierarchical Bayesian formulation that enables exchange of information across groups of related time series, Seeger et al. (2016) proposed a combination of generalized linear models and time series smoothing which is based on a non-Gaussian maximum likelihood estimation, while Chen and Boylan (2008) and Mohammadipour et al. (2012) introduced approaches for grouping the seasonal indices of various products, thus effectively capturing their seasonal patterns. Stimulated by the successful elements of these methods, we investigate the effect of cross-learning modeling in daily demand forecasting, aiming at capturing various dynamics of the examined dataset which are difficult to identify at a series level, but easier to extract at a global level by using properly designed ML methods. This is another contribution of the present study.

3 Methods utilized

In order to properly compare the performance of ML methods to statistical ones, we consider the most popular forecasting methods of each kind, as well as some of their variants that, according to the literature, can lead to better forecasts. For each method, we first describe its elements and then explain how its parameters are determined. Note that most of the methods examined can be trained in an abundance of ways, resulting to different forecasts. Given that it is practically impossible to implement every single one of them, we proceed by adopting the most common ones, that way providing an indication of how the methods utilized could generally perform in practice. In this regard, we discuss where applicable other possible alternatives and reference relevant modifications proposed in past studies to encourage future research.

We should note that the classification of the forecasting methods into statistical and ML is not trivial (Januschowski et al. 2020). Thus, we proceed by adopting the same criterion considered in the study of Makridakis et al. (2018) and the M4 competition, later expressed in a more formal way by Barker (2020). In brief, we consider as statistical any method that prescribes the data generating process, while as ML any method that allows for data relationships to be learned. For example, the forecasting methods that build on exponential smoothing and moving averages are considered as being statistical, while the forecasting methods that build on non-linear regression algorithms, such as NNs and RTs, are considered as being ML.

3.1 Statistical methods

In total, we consider eleven (11) statistical methods: a naive method, a random walk model adjusted for seasonality, three conventional time series methods, the Croston's method and three of its variants/modifications, as well as two temporal aggregation forecasting methods.

- *Naive* The forecasts at time t, $\hat{y}_t$, are equal to the last known observation of the time series, y, as follows:

$$\hat{y}_t = y_{t-1}. \tag{1}$$

  Although very simplistic in nature, Naive has been reported to produce unbiased results, making it a reasonable benchmark (Kourentzes 2013).
- *Seasonal Naive (sNaive)* The forecasts at time t are equal to the last known observation of the same period, $t - m$, as follows:

$$\hat{y}_t = y_{t-m}, \tag{2}$$

  where m is the frequency of the series (e.g., 12 for monthly data). Thus, in contrast to the Naive method, sNaive can capture possible seasonal variations. Although demand time series do not usually display strong seasonality at low cross-sectional and temporal levels, it is still worth investigating this possibility. Note that even if the examined dataset is found to consist of non-seasonal series, the opposite can be true for other datasets, especially when the demand is aggregated at high temporal levels (e.g. monthly and quarterly data).
- *Simple Exponential Smoothing (SES)* The simplest exponential smoothing model, aimed at predicting series without a trend (Gardner 1985). Since SES is applied to the original data directly, potentially involving periods of zero

demand, its performance is expected to deteriorate for series displaying intermittent demand. Yet, it is commonly used in practice (Gardner 2006; Rostami-Tabar et al. 2013). Forecasts are calculated using weighted averages that decrease exponentially across time, specified through the smoothing parameter a as follows:

$$\hat{y}_t = ay_t + (1 - a)\hat{y}_{t-1}. \tag{3}$$

Typically, in an intermittent demand context, low smoothing constant values are recommended in the literature (Syntetos and Boylan 2005; Teunter and Duncan 2009), with a ranging from 0.1 to 0.3. We selected the optimal value from this range by minimizing the in-sample mean squared error (MSE) of the model and initialized it using the first observation of the series. Note that other initializations (e.g., the mean of the series) and ranges for the smoothing parameter (e.g., 0 to 1) were also considered based on suggestions of past studies investigating the optimization and selection of demand forecasting models (Eaves and Kingsman 2004; Boylan et al. 2008; Teunter et al. 2010; Petropoulos et al. 2013; Kourentzes 2014). However, the differences reported in terms of forecasting accuracy and bias in the examined dataset were negligible and, therefore, we did not consider them any further for reasons of brevity.

Note also that since the examined dataset involves a lot of products that are not intermittent, or have low intermittency, in addition to SES we considered ETS (Hyndman et al. 2002), a standard method for automatically forecasting continuous, regular data using exponential smoothing models. However, our results suggested that ETS had a similar performance with SES, while also being more computationally expensive. Therefore, ETS was not consider any further. This can be possibly attributed to the particular characteristics of the examined series, which do not display significant seasonality and trend.

- *Moving Averages (MA)* Moving averages are also used often in practice to forecast demand (Syntetos and Boylan 2005). Forecasts are computed by averaging the last k observations of the series as follows:

$$\hat{y}_t = \frac{\sum_{i=1}^{k} y_{t-i}}{k} \tag{4}$$

In this study, the order of the MA ranges between 2 and 5 and is specified by minimizing the in-sample MSE of the method. In addition, we consider the approach proposed by Svetunkov and Petropoulos (2018), to be named MA-opt, for automatically selecting the optimal order of MA using information criteria.

- *Croston's method (CRO)* Croston (1972), and later Rao (1973), proposed forecasting demand time series by separating them into two components and extrapolating them individually: the non-zero demand size, z_t, and the inter-demand intervals, p_t. The forecasts are given as follows

$$\hat{y}_t = \frac{\hat{z}_t}{\hat{p}_t} \tag{5}$$

and are updated only when demand occurs. Both z_t and p_t are forecasted by SES, originally using a smoothing parameter of 0.1 and an initial value equal to the first observation of each series. Croston's method is regarded as the standard method for forecasting intermittent demand, specifying what the mean demand will be for every future period.

- *Syntetos–Boylan Approximation (SBA)* Syntetos and Boylan (2005) showed that Croston's method is biased. The bias of the method depends on the value of the parameter a used for smoothing the inter-demand intervals. In this regard, they proposed a variant of the Croston's method that utilizes a debiasing factor as follows:

$$\hat{y}_t = \left(1 - \frac{a}{2}\right)\frac{\hat{z}_t}{\hat{p}_t}. \tag{6}$$

As done for the Croston's method, a is set equal to 0.1 and the first observations of z_t and p_t are used for initialization.

- *Shale–Boylan–Johnston Approximation (SBJA)* Shale et al. (2006) proposed another modification to Croston's method for generating unbiased forecasts when the arrival of orders follows a Poisson process, yielding the correction factor when employing either MA or SES. In the first case, the debiasing factor is equal to $(k-1)/k$, while in the second $1 - a/(2-a)$. Our experiments showed that the differences between SBJA and SBA were negligible for the case of SES and, therefore, we decided to report just the ones of the MA for reasons of brevity.
- *Teunter–Syntetos–Babai method (TSB)* Teunter et al. (2011) reported that Croston's method is inappropriate for dealing with obsolescence issues, mainly due to its updating which occurs only in non-zero demand periods. In this respect, they proposed a modification to Croston's method that replaces the inter-demand intervals component with the demand probability, d_t, being 1 if demand occurs at time t and 0 otherwise. Similarly to Croston's method, d_t is forecasted using SES. The forecasts are given as follows

$$\hat{y}_t = \hat{d}_t\hat{z}_t \tag{7}$$

- *Aggregate–Disaggregate Intermittent Demand Approach (ADIDA)* Nikolopoulos et al. (2011) proposed the utilization of temporal aggregation for reducing the presence of zero observations, that way mitigating the undesirable effect of the variance observed in the intervals. In this respect, ADIDA uses equally sized time buckets to perform non-overlapping temporal aggregation and predict the demand over a pre-specified lead time. Various methods can be used for determining the time bucket and extrapolating the aggregated series. In this study we set the time bucket equal to the mean inter-demand interval (Petropoulos and Kourentzes 2015) and use SES to obtain the forecasts.
- *Intermittent Multiple Aggregation Prediction Algorithm (iMAPA)* This method, proposed by Petropoulos and Kourentzes (2015), is another way for implementing temporal aggregation in demand forecasting. However, in contrast to ADIDA, considering a single aggregation level, iMAPA considers multiple ones, aiming at capturing different dynamics of the data (Kourentzes et al. 2014b). Thus, iMAPA proceeds by

averaging the derived point forecasts at each temporal level, generated in this study by SES. The maximum aggregation level is set equal to the maximum inter-demand interval. Note that the default implementation of iMAPA involves selecting between the Croston's method, SBA, and SES, depending on the intermittency and the erraticness of the examined series. We decided not to consider this implementation in order for the results of SES, ADIDA (temporal aggregation), and iMAPA (multiple temporal aggregation) to be directly comparable. However, such an approach could improve forecasting performance.

3.2 Machine Learning methods

In total, we consider seven ML methods: two different implementations of NNs, namely a Multi-Layer Perceptron (MLP) and a Bayesian Neural Network (BNN), two different implementations of Regression Trees (RTs), namely a Random Forest (RF) and a Gradient Boosting Tree (GBT), k-Nearest Neighbour Regression (kNNR), Support Vector Regression (SVR), and Gaussian Processes (GPs).

All methods were trained using the standard approach of constant size, rolling input and output windows (Smyl 2020). That way, the same set of observations used for fitting the statistical models is also utilized for training the ML ones. Since the examined dataset involves daily demand data, spanning from Monday to Saturday, the size of the input window, x_i, is set equal to $n_i \times 6$, with n_i being selected between 1, 2, 3, and 4, in order for the input vector to cover n_i full seasonal periods but avoid unnecessary complexity that larger vectors would have introduced to the training process. The size of the output window, x_o, is set equal to one for three reasons: First, to keep the methods as simple as possible and reduce computational time (Mukhopadhyay et al. 2012), second, to allow computations even for short time series that are common in demand forecasting settings (Lolli et al. 2017), and third, to ensure that the forecasts produced by both ML and statistical methods are directly comparable. In this regard, as done with all statistical methods and proposed for the case of NNs (Kourentzes 2013), the produced one-step-ahead forecasts are used for predicting all h-step-ahead periods, where h is the forecasting horizon. The "optimal" values of the hyper-parameters of the examined ML methods are determined by performing a grid search on a validation set and using the Root Mean Squared Scaled Error (RMSSE) (Makridakis et al. 2020c) as an optimization criterion (for more details see Sect. 4.2).

Due to the nonlinear activation functions used by the ML algorithms, the data are scaled before training between 0 and 1 to avoid computational problems, meet algorithm requirement, and facilitate faster learning (Zhang et al. 1998). The linear transformation, y', is as follows:

$$y'_t = \frac{y_t - y_{min}}{y_{max} - y_{min}}, \tag{8}$$

where y_{min} and y_{max} denote the minimum and maximum value of the training sample, respectively. The reverse transformation can be used to re-scale the forecasts and obtain the final predictions of each method.

- *Multi-Layer Perceptron (MLP)* We constructed a single hidden layer NN using the *RSNNS* R statistical package (Bergmeir and Benítez 2012). The NN consisted of x_i input nodes and $n_h \times x_i$ hidden nodes (`size`), with n_h being selected between 1, 2 and 3, following the practical guidelines of Lippmann (1987) aimed at decreasing the computational time needed for constructing the model (Zhang et al. 1998). We considered the standard backpropagation, Scaled Conjugate Gradient (SCG), and weight decay backpropagation (Møller 1993) for estimating the weights of the network (`learnFunc`), which were initialized randomly. The learning rate was automatically selected between 0.1 and 1, while the maximum iterations (`maxit`) were selected between 100, 250, 500, and 1000. The activation function of the hidden layer (`hiddenActFunc`) was the logistic one, while the activation function of the output layer (`linOut`) was either the logistic or the linear one. In total, 5 MLPs were trained and the median operator was used to average the individual forecasts in order to mitigate possible variations due to poor weight initializations (Kourentzes et al. 2014a).
- *Bayesian Neural Network (BNN)* BNN is similar to the MLP but optimizes the weights according to the Bayesian concept assuming some a priori distributions of errors. The NN was constructed based on the suggestions provided by MacKay (1992) and Dan Foresee and Hagan (1997) and was implemented using the *brnn* R statistical package (Rodriguez and Gianola 2018). The Nguyen and Widrow algorithm (Nguyen and Widrow 1990) was used to assign initial weights and the Gauss–Newton algorithm to perform the optimization, with the μ value used for controlling the optimization process being selected between 0.001, 0.01, and 0.1. The size of the hidden layers (`neurons`) and the number of the training epochs (`epochs`) were the same with those considered for MLP. Accordingly, an ensemble (median) of 5 BNNs was constructed for producing the final forecasts.
- *Random Forest (RF)* RF is a combination of RTs, each one depending on the values of a random vector sampled independently and with the same distribution (Breiman 2001). The accuracy of the method depends on the size of the forest, as well as the strength and correlation of the individual trees. Given that RF averages the predictions of multiple RTs, it is more robust to noise and less likely to over-fit on the training data. We implemented RF using the *randomForest* R statistical package (Liaw and Wiener 2002). The number of non-pruned trees (`ntree`) was selected between 100, 250, 500, and 1000. The minimum size of terminal nodes (`nodesize`) was selected between 5, 10, 100, and 500. The number of variables randomly sampled as candidates at each split (`mtry`) was selected between $x_i/2$, $x_i/3$, $x_i/5$, and $x_i/10$. Bootstrap sampling was done with replacement.
- *Gradient Boosting Trees (GBT)* GBT has principles similar to those of RF, but instead of generating multiple independent trees, it builds one tree at a time, each new tree correcting the errors made by the previously trained one (Freund and Schapire 1997). Since GBT considers more complex data dynamics than RT, it is expected to provide more accurate forecasts (Friedman 2002). However, in contrast to RF, GBT is still susceptible to over-fitting, especially when the data is noisy. We implemented GBT using the *gbm* R statistical package

(Greenwell et al. 2019). To allow for slow learning and better generalization, we selected the learning rate (`shrinkage`) between 0.001, 0.01, and 0.1, and the maximum tree depth between 1, 2, 4, 8, and 16. The total number of trees considered (`n.trees`) was selected between 100, 250, 500, and 1000. The distribution used to fit the model (`distribution`) was either the Gaussian or the Laplace one.

- *K-Nearest Neighbor Regression (KNNR)* KNNR is a similarity-based method, generating forecasts according to the Euclidean distance computed between the points used for training and testing. Given x_i points as inputs, the method picks the closest k points of the training sample to them and then sets the prediction equal to the average of their corresponding target values. We implemented the method using the *caret* R statistical package (Kuhn 2018). `k` was selected between 3 and 99 with a step of 3.
- *Support Vector Regression (SVR)* SVR generates forecasts by identifying the hyperplane that maximizes the margin between two classes and minimizes the total error under tolerance (Schölkopf and Smola 2001). In order to reduce complexity and accelerate computations, we considered ν-regression which constructs few support vectors with respect to the total number of samples in the dataset. We implemented the method using the *e1071* R statistical package (Meyer et al. 2019). The kernel used for training and predicting (`kernel`) was selected between the linear, polynomial, radial basis, and sigmoid ones. The tolerance of termination criterion (`tolerance`) was selected between 0.001, 0.01, and 0.1, while the ν value (`nu`) ranged between 0.3 and 0.7.
- *Gaussian Processes (GP)* GP associates the dependent variable with multiple normally distributed random variables so that their combination replicates the target as much as possible (Rasmussen and Williams 2006). The combination is based on the similarity of the points and is performed using a kernel function. The method was implemented using the *kernlab* R statistical package (Karatzoglou et al. 2004). We considered various kernel functions (`kernel`) for training and predicting (radial basis, polynomial, linear, hyperbolic tangent, Laplacian, Bessel, and ANOVA radial basis), with their parameters being automatically selected. The initial noise variance (`var`) was selected between 0.001, 0.01, and 0.1.

In addition to the ML methods described above, trained in a series-by-series fashion, we also examine their cross-learning counter-parts, as discussed in Sect. 2, implemented as follows:

- Each time series is individually scaled from 0 to 1 according to the transformation of Eq. 8.
- For each series we create the corresponding constant size input and output windows. Then, for each series we randomly select 3 instances and use them to build the training sample of the model. We do not consider every possible rolling window as done for the models trained in a series-by-series fashion since, by doing so, the time required for training would have increased substantially.

- The examined ML method is trained using as input the windows created earlier for the complete dataset. The settings and hyper-parameters are selected as described in Sect. 3.2.
- For each series we generate h-step-ahead forecasts by providing the ML method trained previously with its last x_i observations (respective input window).
- The forecasts are re-scaled using the inverse transformation formula.

Since the literature in demand forecasting suggests the utilization of time series features as regressor variables in ML forecasting methods (Gutierrez et al. 2008; Nasiri Pour 2008), a suggestion verified recently through the results of the M4 competition for the case of continuous, conventional series (Montero-Manso et al. 2020), we also consider including two features as input to our cross-learning models in addition to the x_i historical observations described earlier. These features are the coefficient of variation of non-zero demands (*CV*2) and the average number of time periods between two successive non-zero demands (*ADI*), as proposed by Syntetos and Boylan (2005). The assumption made by this approach is that the series that display similarities, either in terms of intermittency or erraticness, will require a similar processing from the forecasting model used (Petropoulos et al. 2014; Spiliotis et al. 2020). Thus, the incorporation of these variables could enhance cross-learning and facilitate pattern recognition for series that display different characteristics. Note that *CV*2 has been examined by Abolghasemi et al. (2020), concluding that different models are appropriate for series displaying different coefficients of variation. The length of the series was also considered as a potentially useful feature (Petropoulos et al. 2014). However, since the inclusion of time series length led to slightly worse results, we decided to exclude it from the analysis. The fact that the examined dataset involves series of mostly the same lengths may justify this last finding.

4 Empirical evaluation

4.1 Dataset

The eighteen (18) methods described in Sect. 3 are evaluated on 3300 real time series of various consumption goods sold by a major retailer in the region of Attica, Greece, including cheese and dairy products, meat, fruit, sweets, pasta and rice, meals and snacks, frozen goods, wine, coffee, beverages and non-alcoholic drinks, self-care items, cleaning equipment, and other. Each series represents the daily demand of an SKU, spanning from Monday to Saturday. The dataset covers a full calendar year of demand (2017), but the length of the series varies from 175 to 311 days, depending on the time each product was introduced to the market. The vast majority of the series included more than 280 observations.

Following the suggestions of Syntetos and Boylan (2005), for each series we first compute the CV^2 (squared coefficient of variation of the demand when it occurs) and *ADI* (average inter-demand interval) values, and then use the thresholds proposed by Syntetos et al. (2005) to categorize them (0.5 and 4/3, respectively). CV^2 represents demand size erraticness, while *ADI* intermittency, thus allowing an intuitive categorization of the data. Figure 1 presents the 3300 series in a *CV*^{2}-*ADI* scatter-plot. In total, the dataset includes 240 intermittent, 470 lumpy, 1326 erratic, and 1264 smooth series. Note that a considerable amount of series implies limited intermittency (about 78%), mainly due to the high geographical level where the demand is reported and the nature of the dataset which involves many fast-moving, essential consumption goods. On the other hand, many series of extreme erraticness exist (about 54%), making their extrapolation a challenging task. Figure 2 provides four representative examples of the time series in the dataset, one for each category.

At this point we should note that the particular characteristics of the examined dataset may significantly affect the conclusions drawn in the present study. As noted, most of the series are non-seasonal, display low intermittency, and cover a single calendar year. As a result, methods that are capable of capturing seasonality (at weekly and/or yearly level) and dealing with intermittency, may not display their

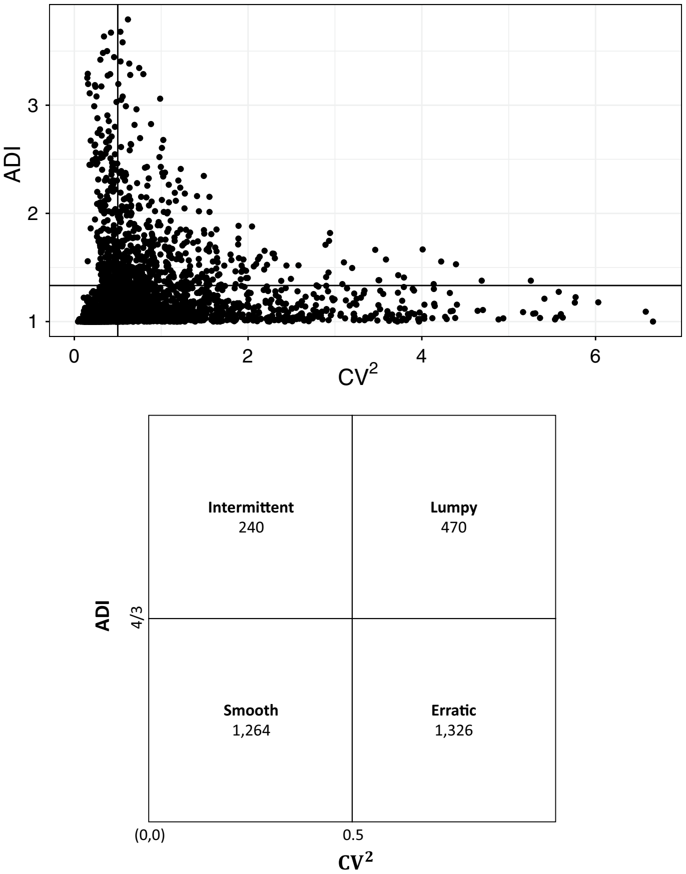

Fig. 1 Demand classification of series based on their intermittency (*ADI*) and erraticness (*CV*^{2}). The graph on the top presents the 3300 series of the dataset in a *ADI*-*CV*^{2} scatter-plot, while the graph at the bottom the population of the four discrete categories. In total, the dataset includes 240 intermittent, 470 lumpy, 1326 erratic, and 1264 smooth series

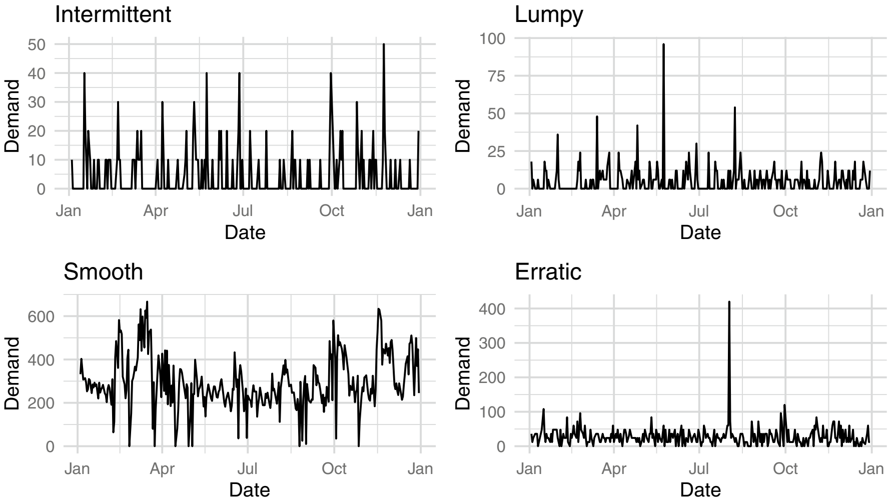

Fig. 2 Example time series. From left to right and from top to bottom, an intermittent, a lumpy, a smooth, and an erratic demand time series is presented

complete potential. In addition, given that the dataset does not include explanatory variables (e.g., product prices, promotions, and weather conditions) and information about the hierarchy of the products, cross-learning methods will omit useful information that could be used to facilitate learning and enhance their forecasting performance.

4.2 Experimental design

Typically, daily SKU demand data, like the one examined in the present case-study, do not display strong seasonal patterns. However, seasonal effects, changing the level of the demand, may become evident across the year, e.g. for different months, seasons or quarters. In order to properly evaluate the performance of the methods utilized, we proceed by assessing their accuracy and bias in four different periods across the year. To do so, we consider a forecasting horizon of four weeks (24-step-ahead forecasts) and generate our first forecasts in the middle of September to predict the demand for the following 24 working days. Then, we reveal the actual demand of the forecasted period and repeat this experiment to forecast the following four weeks and so on till the end of the calendar year. This is equivalent to a rolling-origin evaluation of a window of 24 (Tashman 2000). The forecasting horizon was determined based on the lead time adopted by the company, being also similar to the one considered recently in the M5 forecasting competition whose aim is to accurately predict the daily units sales of 3049 products sold by Walmart in ten of its US stores (Makridakis et al. 2020c).

Note that the values of the hyper-parameters of the examined ML methods were determined by performing a grid search on a validation set which consisted of the last 24 observations of the first training sample of the experiment. The values of the hyper-parameters were specified as described in Sect. 3.2.

Measuring forecasting performance for multiple demand time series that may involve zeroes is not trivial (Davydenko and Fildes 2013). Although standard error measures, like the Mean Absolute Error (MAE), can effectively measure forecasting accuracy for each series separately, they fail to summarize results across multiple series given that they are scale dependent. Percentage errors, like the Mean Absolute Percentage Error (MAPE), deal with this problem in an intuitive way, but are inappropriate for intermittent demand data due to the zero values present. Finally, relative errors, like the Geometric Mean Relative Absolute Error (GMRAE), although intuitive, depend on the computation of a benchmark model (Fildes 1992). Given the nature of the series, involving only integer demand values, it is possible for the benchmark to achieve zero error, making the estimation of a geometric mean impossible.

Taking the above into consideration, we proceed by assessing the performance of the methods utilized both in terms of accuracy and bias, using the Root Mean Squared Scaled Error (RMSSE) and the Absolute Mean Scaled Error (AMSE), respectively, as follows:

$$RMSSE = \sqrt{\frac{1}{h}\frac{\sum_{t=n+1}^{n+h}(y_t - \hat{y}_t)^2}{\frac{1}{n-1}\sum_{t=2}^{n}(y_t - y_{t-1})^2}}, \tag{9}$$

$$AMSE = \frac{1}{h}\frac{\left|\sum_{t=n+1}^{n+h} y_t - \hat{y}_t\right|}{\frac{1}{n-1}\sum_{t=2}^{n}\left|y_t - y_{t-1}\right|}. \tag{10}$$

Both measures are variants of the Mean Absolute Scaled Error (MASE), proposed by Hyndman and Koehler (2006), which is widely accepted in the forecasting literature (Franses 2016). RMSSE is a scaled version of the root mean squared error, while AMSE of the mean error, with their scaling being the forecast error of the Naive method. Lower RMSSE and AMSE values are better. Makridakis et al. (2020c) also use RMSSE to evaluate the point forecasts of the submitting entries for the M5 forecasting competition.

Note that squared error is preferred over absolute error for measuring forecasting accuracy, as the former measure is minimised by the mean demand value, while the latter by the median (Schwertman et al. 1990). This is critical as the examined series involve a lot of zero values, meaning that minimizing the error by the median would possibly lead to undershooting the demand (Kolassa 2016). Moreover, optimizing MSE leads to less biased forecasts.

The results of the empirical evaluation are summarized by averaging the RMSSE and AMSE values computed per series and assessment period. Thus, each method is evaluated using a sample of 3300 series × 4 assessment periods × 24 days = 316,800 point forecasts.

5 Results and discussion

Table 1 presents the performance of the eleven (11) statistical methods considered in the case-study, along with that of the seven (7) ML methods, trained in a series-by-series fashion. The results are summarized for the RMSSE and AMSE measures separately to evaluate the different methods both in terms of bias and accuracy. The fourth and fifth columns of the table display the ranks of the methods per measure, while the following two columns their percentage improvement, if any, over the Croston's method. The last column reports the computational time (seconds) required for predicting all 3300 series using a server of the following characteristics: 8 cores, 16 GB RAM, 500 GB, HDD, Windows 10.

Observe that according to Table 1, the top-four performing methods in terms of AMSE and RMSSE are ML ones, namely the GPT, RF, SVR, and the KNNR. Moreover, GP is outperformed only by SBA, followed by CRO. On the contrary, the two NN methods are outperformed by the statistical ones, although not by all of them.

Table 1 Average performance of the statistical and ML methods (trained in a series-by-series fashion) considered in terms of bias (AMSE) and accuracy (RMSSE) for the complete dataset of 3300 series

Method	Performance		Rank		% Improvement over Croston		Computational time (s)
	AMSE	RMSSE	AMSE	RMSSE	AMSE	RMSSE	
Statistical methods							
Naive	1.141	1.236	18	17	− 47.7	− 14.5	0.40
sNaive	1.047	1.457	16	18	− 35.5	− 34.9	0.65
SES	0.881	1.127	14	13	− 14.1	− 4.4	67.55
MA	1.065	1.200	17	16	− 37.9	− 11.1	70.35
MA-opt	0.877	1.132	12	14	− 13.5	− 4.8	6995.34
CRO	0.773	1.080	7	7	0.0	0.0	14.30
SBA	0.757	1.069	5	5	2.1	1.0	14.25
SBJA	1.022	1.174	15	15	− 32.3	− 8.7	92.08
TSB	0.867	1.122	11	11	− 12.2	− 3.9	265.08
ADIDA	0.817	1.101	8	9	− 5.7	− 1.9	58.86
iMAPA	0.843	1.111	10	10	− 9.1	− 2.9	133.07
ML methods (series-by-series)							
MLP	0.817	1.095	9	8	− 5.8	− 1.4	1026.27
BNN	0.877	1.123	13	12	− 13.6	− 4.0	19328.62
RF	0.660	1.033	2	2	14.6	4.4	489.81
GBT	0.648	1.026	1	1	16.1	5.0	131.63
KNNR	0.684	1.036	4	4	11.5	4.1	21.58
SVR	0.670	1.036	3	3	13.3	4.1	90.86
GP	0.760	1.073	6	6	1.7	0.7	234.17

The ranks and the percentage improvements of the methods over the Croston's method are also reported per measure. The computational time required in seconds for predicting all 3300 series is also provided

This finding indicates that particular ML methods are capable of producing less biased and more accurate forecasts than well-established, statistical ones, confirming the great potential of the former and illustrating a fertile area for future research that focuses on various ML methods other than the NN ones typically examined in the literature.

Note also that the results are consistent across the two measures used, meaning that the more accurate methods tend to be less biased too. On the other hand, the improvements reported for the two measures over CRO do differ in terms of margin. For example, GBT, the best-performing method of Table 1, improves forecasting accuracy by 5%, but reduces the bias by more than 16%. Accordingly, the top-four performing ML methods are found to improve RMSSE on average by 4.4% and AMSE by 14%. Given that bias is usually more important than accuracy in inventory control settings (Kourentzes 2013), the potential benefits of utilizing such ML forecasting methods could be substantial.

Our results confirm among others the superiority of the advanced statistical methods over the standard ones, exactly as reported in the literature. For example, SBA performs better than CRO, while MA-opt and SBJA better than MA. In addition, the two methods that consider temporal aggregation are among the most accurate and unbiased statistical forecasting methods. Finally, it is verified that conventional time series methods like SES, MA, Naive, and sNaive are inappropriate for predicting irregular demand data.

Table 2 summarizes the performance of the ML methods when trained in cross-learning fashion instead of a series-by-series one. As described in Sect. 3.2, two different implementations of cross-learning modelling are considered. In the first case, D, the historical observations are used for training the model, while in the second, F, two features (*CV*2 and *ADI*) are used in compliance to the historical observations to facilitate learning. For each method and modelling approach, percentage improvements are reported both over the Croston's method and its corresponding series-by-series implementation.

Observe that in all cases apart from the NNs, cross-learning leads to less accurate and more biased forecasts than the series-by-series modeling approach. However, the accuracy of MLP and BNN methods is improved by about 3% and 5%, respectively. Similarly, the bias of the MLP and the BNN methods is improved by about 7% and 14%, respectively. Observe also that the improvements are greater when features are used as regressor variables, making the two NN methods outperform the Croston's method, which was not previously the case.

Our results suggest that extracting information from multiple series to forecast the individual ones is a beneficial strategy for NNs, although an inappropriate one for other types of ML methods, such as support vector machines and RTs. This can be attributed to the particularities of the ML methods used, their learning capacity, and complexity. For example, NNs are complicated in nature and require the estimation of numerous weights. When data is limited, the values of these weights cannot be properly estimated and, therefore, cross-learning becomes beneficial. On the other hand, RTs and KNNR are robust to over-fitting and good at dealing with sparse, noisy data. Thus, we conclude that although cross-learning displays some potential, its benefits are more likely to be leveraged by particular types of ML methods.

Table 2 Comparison of ML methods trained in a series-by-series (SbS) fashion to those utilizing cross-learning (CL), either by using historical data alone (D) or additional time series features as regressor variables (F)

Method	Modelling approach	Performance		% Improvement of CL over SbS		% Improvement over Croston		Computational time (s)
		AMSE	RMSSE	AMSE	RMSSE	AMSE	RMSSE	
MLP	D	0.778	1.076	4.8	1.8	− 0.6	0.4	389.89
	F	0.760	1.067	7.0	2.5	1.7	1.2	541.98
BNN	D	0.775	1.077	11.7	4.1	− 0.3	0.3	478.10
	F	0.751	1.066	14.4	5.1	2.8	1.3	1699.40
RF	D	0.706	1.048	− 7.1	− 1.5	8.6	2.9	64.30
	F	0.681	1.037	− 3.2	− 0.5	11.9	4.0	53.33
GBT	D	0.719	1.046	− 10.9	− 1.9	7.0	3.2	34.69
	F	0.716	1.052	− 10.5	− 2.5	7.3	2.6	34.91
KNNR	D	0.726	1.052	− 6.2	− 1.6	6.0	2.6	27.40
	F	0.730	1.055	− 6.7	− 1.8	5.6	2.4	27.77
SVR	D	0.765	1.070	− 14.2	− 3.3	1.0	0.9	33.41
	F	0.748	1.059	− 11.7	− 2.3	3.1	1.9	32.14
GP	D	0.795	1.082	− 4.6	− 0.8	− 2.9	− 0.2	53.71
	F	0.798	1.083	− 5.1	− 0.9	− 3.3	− 0.2	53.32

The comparison is done in terms of bias (AMSE) and accuracy (RMSSE) for the complete dataset of 3300 series. The percentage improvements of the cross-learning methods over the Croston's method, as well as their series-by-series equivalents, are reported per measure. The computational time required in seconds for predicting all 3300 series is also provided

Regarding the trade-off between forecasting performance and computational cost, Table 1 suggests that the average computational time of the statistical and ML methods is about 12 min and 51 min, respectively. Our results indicate that when ML methods are trained in a series-by-series fashion, a significant cost must be paid to improve forecasting performance, even to a small extent. Taking into consideration that fast, sub-optimal forecasts may sometimes be preferable to slow, "optimal" ones (Nikolopoulos and Petropoulos 2018), we argue that companies and organizations should carefully design their forecasting processes by quantifying monetary savings as a result of sub-optimal solutions. Nevertheless, according to Table 2, the average computational time of the cross-leaning ML methods drops to about 4 min, i.e., becomes less than the one of the statistical methods. Given that cross-learning methods were found to perform better or similarly well with their series-by-series counterparts, we conclude that cross-learning can effectively improve forecasting performance, while also reducing computational costs. This finding supports the exploitation of cross-learning approaches and highlights their potential, especially in the big data era.

At this point we should note that the results reported in Tables 1 and 2 focus on the average performance of the examined methods, meaning that different methods could possibly be appropriate for series of different characteristics. Moreover, they do not provide any indication as to whether the achieved improvements by the

ML methods are statistically significant. In this respect, we apply the Multiple Comparisons with the Best (MCB) test that compares whether the average ranking of a forecasting method is significantly different than the others (Koning et al. 2005). If the confidence intervals of two methods overlap, their ranked performances are not statistically different, and vice versa. We perform such a test both for the AMSE and the RMSSE measures, considering the whole dataset or subsets of it based on the categorization of the series into intermittent, lumpy, smooth, and erratic. The best implementation of each ML method is considered for the analysis.

Figure 3 presents the results of the MCB test for the complete dataset of 3300 series for each performance measure separately. Observe that RF and GBT perform significantly better than the rest of the forecasting methods examined, both in terms of AMSE and RMSSE, being the top-ranked ones, exactly as suggested by Table 1. RF and GBT are followed by the remaining five ML methods, outperforming all the statistical benchmarks considered, apart from SBA which is ranked 7th. Observe also that the MCB ranks of the examined forecasting methods are the same across the two measures and in alignment with those reported in Tables 1 and 2. In this regard, it is confirmed that the results and the conclusions drawn earlier are statistically significant.

Interesting conclusions can also be made when examining the results of the MCB test per time series categories. The results are presented in Figs. 4 and 5 for the case of the RMSSE and the AMSE measures, respectively. We observe that the intermittency plays a pivotal role in the performance of the forecasting methods used, with SBA, a statistical method, not being statistically different than the

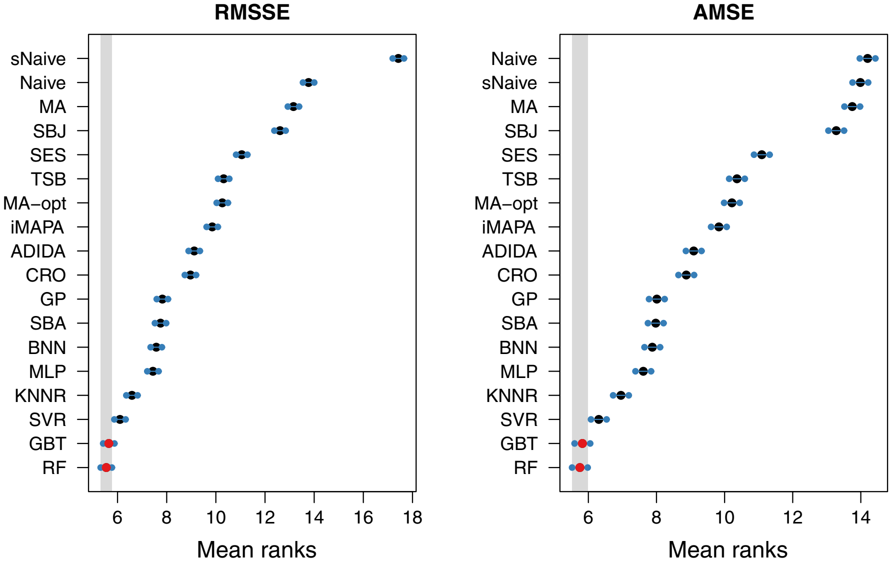

Fig. 3 MCB significance tests for the statistical and the cross-learning ML forecasting methods considered. The results are presented for the complete dataset of 3300 time series, both in terms of accuracy (RMSSE) and bias (AMSE)

top-performing ML ones for the case of intermittent and lumpy series. On the contrary, when intermittency is limited, ML methods, and particularly the RT-based ones, are significantly better. Thus, we conclude that although ML methods are more appropriate for predicting daily SKU demand, some statistical methods may perform similar for intermittent and lumpy series.

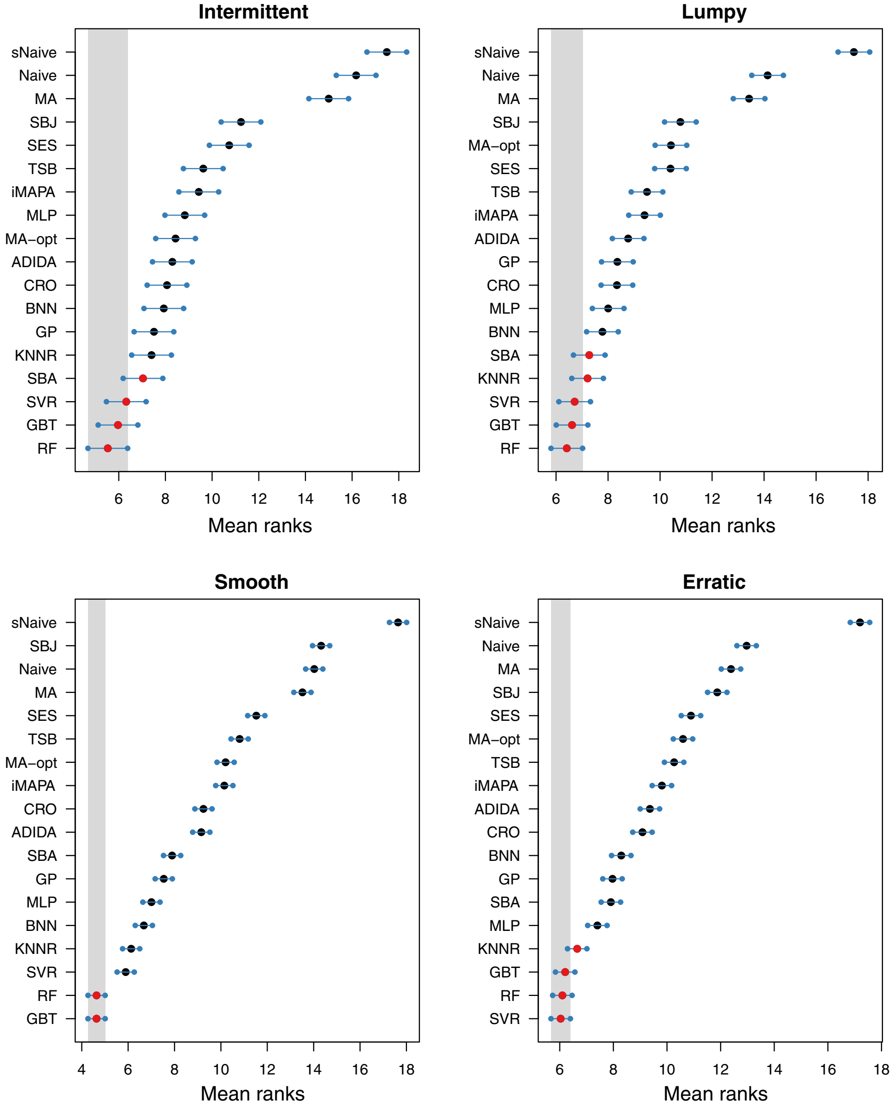

Fig. 4 MCB significance tests for the statistical and the cross-learning ML forecasting methods considered. The results are presented for intermittent, lumpy, smooth, and erratic demand series separately. The ranks are computed according to RMSSE

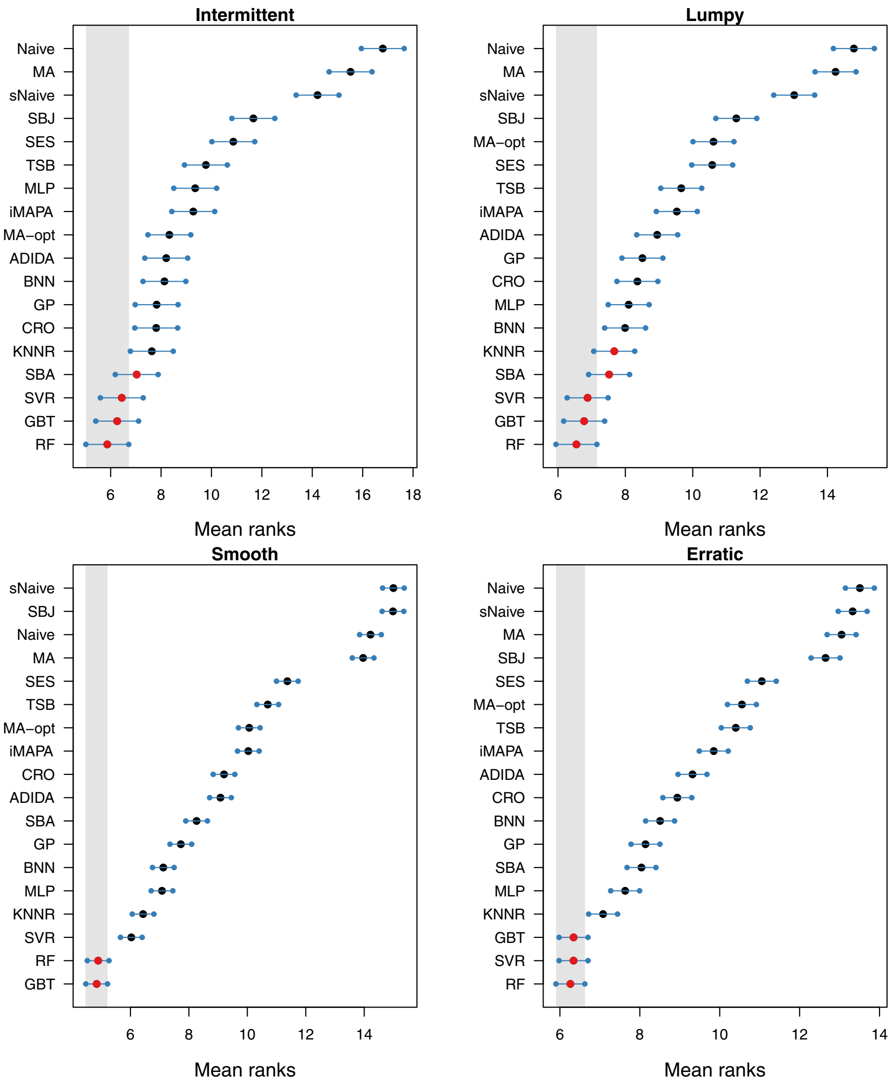

Fig. 5 MCB significance tests for the statistical and the cross-learning ML forecasting methods considered. The results are presented for intermittent, lumpy, smooth, and erratic demand series separately. The ranks are computed according to AMSE

6 Conclusions

ML has been proven to be an effective solution for improving the performance of statistical forecasting methods in continuous, regular time series forecasting. However, limited research has been conducted in the area of demand forecasting, with studies comparing ML approaches with statistical ones being inconclusive with regards to the superiority of the one over the other. Furthermore, most of the studies have focused on the utilization of NNs, a special type of ML methods, thus ignoring the potential advantages of other alternatives. Limited test samples and a lack of a sensible benchmark further restrain the extraction of reliable and generalized conclusions.

This study has shed some light on this area of research by (1) evaluating the performance of ML methods for the case of daily SKU demand forecasting using a set of sensible statistical benchmarks, both simple and advanced ones, (2) considering a variety of ML methods and not just NNs, (3) exploring the potential of cross-learning that enables modeling across multiple series, and (4) identifying the most suitable forecasting methods for different types of series in terms of intermittency and erraticness.

Our results show that ML methods can provide significantly less biased and more accurate forecasts than well-established, statistical methods, like the Croston's method and its variants. In addition, it is shown that cross-learning can improve the forecasting performance of NNs, trained in a series-by-series fashion, that way outperforming standard statistical benchmarks. This is particularly true when time series features are used in addition to historical data to train the networks. However, cross-learning had a negative impact for the rest of the ML methods examined, indicating that different modeling approaches should be used based on the particularities of the forecasting method. Moreover, training ML methods in a cross-learning fashion can lead to reduced computational costs.

Although this study supports the utilization of ML methods in demand forecasting and proposes some ways for improving their performance, some questions remain unanswered. For example, it is still unclear why some ML models, trained in a series-by-series fashion, perform better than others and why particular types of ML methods are more effective in applying cross-learning. Moreover, since intermittency and erraticness seem to affect to some extent the performance of the forecasting methods, it would be interesting to further investigate their response when the values of these two features vary. In addition, given that this study explores only two basic time series features to facilitate cross-learning and assist the methods to identify series of similar characteristics, it would be reasonable to examine additional, explanatory features, such as data autocorrelation and normality. Other limitations of the present study refer to the particular characteristics of the examined dataset: Most of the series were non-seasonal, displayed low intermittency, and covered a single calendar year. Explanatory variables and information about the hierarchy of the products, which could be used to enhance cross-learning, was also missing. Thus, future research could expand this analysis for richer datasets, investigating further the benefits of cross-learning methods.

References

Abolghasemi M, Beh E, Tarr G, Gerlach R (2020) Demand forecasting in supply chain: the impact of demand volatility in the presence of promotion. Comput Ind Eng 142:106380

Ali ÖG, Sayın S, van Woensel T, Fransoo J (2009) SKU demand forecasting in the presence of promotions. Expert Syst Appl 36:12340–12348

Babai M, Dallery Y, Boubaker S, Kalai R (2019) A new method to forecast intermittent demand in the presence of inventory obsolescence. Int J Prod Econ 209:30–41

Barker J (2020) Machine learning in M4: what makes a good unstructured model? Int J Forecast 36:150–155

Bergmeir C, Benítez JM (2012) Neural networks in R using the stuttgart neural network simulator: RSNNS. J Stat Softw 46:1–26

Bojer CS, Meldgaard JP (2020) Kaggle forecasting competitions: an overlooked learning opportunity. Int J Forecast. https://doi.org/10.1016/j.ijforecast.2020.07.007

Boutselis P, McNaught K (2019) Using Bayesian networks to forecast spares demand from equipment failures in a changing service logistics context. Int J Prod Econ 209:325–333

Boylan JE, Syntetos AA (2009) Spare parts management: a review of forecasting research and extensions. IMA J Manag Math 21:227–237

Boylan JE, Syntetos AA, Karakostas GC (2008) Classification for forecasting and stock control: a case study. J Oper Res Soc 59:473–481

Breiman L (2001) Random forests. Mach Learn 45:5–32

Brown RG (1959) Statistical forecasting for inventory control. McGraw-Hill, New York

Carmo JL, Rodrigues AJ (2004) Adaptive forecasting of irregular demand processes. Eng Appl Artif Intell 17:137–143

Chapados N (2014) Effective Bayesian modeling of groups of related count time series. In: Xing EP, Jebara T (eds) Proceedings of the 31st international conference on machine learning. PMLR volume 32 of proceedings of machine learning research, Bejing, China, pp 1395–1403

Chen H, Boylan JE (2008) Empirical evidence on individual, group and shrinkage seasonal indices. Int J Forecast 24:525–534

Croston JD (1972) Forecasting and stock control for intermittent demands. J Oper Res Soc 23:289–303

Dan Foresee F, Hagan MT (1997) Gauss–Newton approximation to Bayesian learning. In: IEEE international conference on neural networks-conference proceedings, vol 3, pp 1930–1935

Davydenko A, Fildes R (2013) Measuring forecasting accuracy: the case of judgmental adjustments to SKU-level demand forecasts. Int J Forecast 29:510–522

Eaves AHC, Kingsman BG (2004) Forecasting for the ordering and stock-holding of spare parts. J Oper Res Soc 55:431–437

Fildes R (1992) The evaluation of extrapolative forecasting methods. Int J Forecast 8:81–98

Franses PH (2016) A note on the mean absolute scaled error. Int J Forecast 32:20–22

Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55:119–139

Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38:367–378

Gardner ES Jr (1985) Exponential smoothing: the state of the art. J Forecast 4:1–28

Gardner ES (2006) Exponential smoothing: the state of the art part II. Int J Forecast 22:637–666

Ghobbar AA, Friend CH (2003) Evaluation of forecasting methods for intermittent parts demand in the field of aviation: a predictive model. Comput Oper Res 30:2097–2114

Greenwell B, Boehmke B, Cunningham J, Developers G (2019) gbm: Generalized Boosted Regression Models. R package version 2.1.5

Gutierrez RS, Solis AO, Mukhopadhyay S (2008) Lumpy demand forecasting using neural networks. Int J Prod Econ 111:409–420

Hasni M, Aguir M, Babai M, Jemai Z (2019) On the performance of adjusted bootstrapping methods for intermittent demand forecasting. Int J Prod Econ 216:145–153

Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2:359–366

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

Hyndman RJ, Koehler AB, Snyder RD, Grose S (2002) A state space framework for automatic forecasting using exponential smoothing methods. Int J Forecast 18:439–454

Januschowski T, Gasthaus J, Wang Y, Salinas D, Flunkert V, Bohlke-Schneider M, Callot L (2020) Criteria for classifying forecasting methods. Int J Forecast 36:167–177

Johnston FR, Boylan JE, Shale EA (2003) An examination of the size of orders from customers, their characterisation and the implications for inventory control of slow moving items. J Oper Res Soc 54:833–837

Karatzoglou A, Smola A, Hornik K, Zeileis A (2004) kernlab: an S4 package for kernel methods in R. J Stat Softw 11:1–20

Kolassa S (2016) Evaluating predictive count data distributions in retail sales forecasting. Int J Forecast 32:788–803

Koning AJ, Franses PH, Hibon M, Stekler HO (2005) The M3 competition: statistical tests of the results. Int J Forecast 21:397–409

Kourentzes N (2013) Intermittent demand forecasts with neural networks. Int J Prod Econ 143:198–206

Kourentzes N (2014) On intermittent demand model optimisation and selection. Int J Prod Econ 156:180–190

Kourentzes N, Barrow DK, Crone SF (2014a) Neural network ensemble operators for time series forecasting. Expert Syst Appl 41:4235–4244

Kourentzes N, Petropoulos F, Trapero JR (2014b) Improving forecasting by estimating time series structural components across multiple frequencies. Int J Forecast 30:291–302

Kuhn M (2018) caret: Classification and Regression Training. R package version 6.0-81

Liaw A, Wiener M (2002) Classification and regression by randomforest. R News 2:18–22

Lippmann RP (1987) An introduction to computing with neural nets. IEEE ASSP Mag 4:4–22

Lolli F, Gamberini R, Regattieri A, Balugani E, Gatos T, Gucci S (2017) Single-hidden layer neural networks for forecasting intermittent demand. Int J Prod Econ 183:116–128

MacKay DJC (1992) Bayesian interpolation. Neural Comput 4:415–447

Makridakis S, Spiliotis E, Assimakopoulos V (2018) Statistical and machine learning forecasting methods: concerns and ways forward. PLoS ONE 13:1–26

Makridakis S, Hyndman RJ, Petropoulos F (2020a) Forecasting in social settings: the state of the art. Int J Forecast 36:15–28

Makridakis S, Spiliotis E, Assimakopoulos V (2020b) The M4 competition: 100,000 time series and 61 forecasting methods. Int J Forecast 36:54–74

Makridakis S, Spiliotis E, Assimakopoulos V (2020c) The M5 competition: competitors guide. https://mofc.unic.ac.cy/m5-competition/. Accessed 01 Sept 2020

Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F (2019) e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-1

Mohammadipour M, Boylan J, Syntetos A (2012) The application of product-group seasonal indexes to individual products. Foresight Int J Appl Forecast 26:20–26

Møller MF (1993) A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw 6:525–533

Montero-Manso P, Athanasopoulos G, Hyndman RJ, Talagala TS (2020) FFORMA: feature-based forecast model averaging. Int J Forecast 36:86–92

Mukhopadhyay S, Solis AO, Gutierrez RS (2012) The accuracy of non-traditional versus traditional methods of forecasting lumpy demand. J Forecast 31:721–735

Nasiri Pour AA, Rostami Tabar B, Rahimzadeh A (2008) A hybrid neural network and traditional approach for forecasting lumpy demand. World Academy of Science, Engineering and Technology, Paris

Nguyen D, Widrow B (1990) Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. IJCNN Int Joint Conf Neural Netw 13:C21

Nikolopoulos K, Petropoulos F (2018) Forecasting for big data: does suboptimality matter? Comput Oper Res 98:322–329

Nikolopoulos K, Syntetos AA, Boylan JE, Petropoulos F, Assimakopoulos V (2011) An aggregate-disaggregate intermittent demand approach (ADIDA) to forecasting: an empirical proposition and analysis. J Oper Res Soc 62:544–554

Nikolopoulos KI, Babai MZ, Bozos K (2016) Forecasting supply chain sporadic demand with nearest neighbor approaches. Int J Prod Econ 177:139–148

Petropoulos F, Kourentzes N (2015) Forecast combinations for intermittent demand. J Oper Res Soc 66:914–924

Petropoulos F, Nikolopoulos K, Spithourakis G, Assimakopoulos V (2013) Empirical heuristics for improving intermittent demand forecasting. Ind Manag Data Syst 113:683–696

Petropoulos F, Makridakis S, Assimakopoulos V, Nikolopoulos K (2014) Horses for courses in demand forecasting. Eur J Oper Res 237:152–163

Pooya A, Pakdaman M, Tadj L (2019) Exact and approximate solution for optimal inventory control of two-stock with reworking and forecasting of demand. Oper Res Int J 19:333–346

Rao AV (1973) A comment on: Forecasting and stock control for intermittent demands. J Oper Res Soc 24:639–640

Rasmussen CE, Williams C (2006) Gaussian processes for machine learning. The MIT Press, Cambridge

Rodriguez PP, Gianola D (2018) brnn: Bayesian Regularization for Feed-Forward Neural Networks. R package version 7

Rostami-Tabar B, Babai MZ, Syntetos A, Ducq Y (2013) Demand forecasting by temporal aggregation. Naval Res Logist (NRL) 60:479–498

Salinas D, Flunkert V, Gasthaus J, Januschowski T (2020) DeepAR: probabilistic forecasting with autoregressive recurrent networks. Int J Forecast 36:1181–1191

Schölkopf B, Smola AJ (2001) Learning with kernel: support vector machines, regularization, optimization and beyond. The MIT Press, Cambridge

Schwertman NC, Gilks AJ, Cameron J (1990) A simple noncalculus proof that the median minimizes the sum of the absolute deviations. Am Stat 44:38–39

Seaman B (2018) Considerations of a retail forecasting practitioner. Int J Forecast 34:822–829

Seeger MW, Salinas D, Flunkert V (2016) Bayesian intermittent demand forecasting for large inventories. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R (eds) Advances in neural information processing systems, vol 29. Curran Associates Inc, Red Hook, pp 4646–4654

Shale EA, Boylan JE, Johnston FR (2006) Forecasting for intermittent demand: the estimation of an unbiased average. J Oper Res Soc 57:588–592

Smyl S (2020) A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. Int J Forecast 36:75–85

Spiliotis E, Kouloumos A, Assimakopoulos V, Makridakis S (2020) Are forecasting competitions data representative of the reality? Int J Forecast 36:37–53

Spithourakis GP, Petropoulos F, Babai MZ, Nikolopoulos K, Assimakopoulos V (2011) Improving the performance of popular supply chain forecasting techniques. Supply Chain Forum Int J 12:16–25

Svetunkov I, Petropoulos F (2018) Old dog, new tricks: a modelling view of simple moving averages. Int J Prod Res 56:6034–6047

Syntetos AA, Boylan JE (2005) The accuracy of intermittent demand estimates. Int J Forecast 21:303–314

Syntetos AA, Boylan JE, Croston JD (2005) On the categorization of demand patterns. J Oper Res Soc 56:495–503

Syntetos AA, Nikolopoulos K, Boylan JE (2010) Judging the judges through accuracy-implication metrics: the case of inventory forecasting. Int J Forecast 26:134–143

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

Teunter RH, Duncan L (2009) Forecasting intermittent demand: a comparative study. J Oper Res Soc 60:321–329

Teunter R, Syntetos A, Babai M (2010) Determining order-up-to levels under periodic review for compound binomial (intermittent) demand. Eur J Oper Res 203:619–624

Teunter RH, Syntetos AA, Babai MZ (2011) Intermittent demand: linking forecasting to inventory obsolescence. Eur J Oper Res 214:606–615

Willemain TR, Smart CN, Schwarz HF (2004) A new approach to forecasting intermittent demand for service parts inventories. Int J Forecast 20:375–387

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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.