Int. J. Production Economics 231 (2021) 107837

Contents lists available at ScienceDirect

International Journal of Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Forecasting of customer demands for production planning by local k-nearest neighbor models

Mirko Kück [a,*], Michael Freitag [a,b]

[a] University of Bremen, Faculty of Production Engineering, Badgasteiner Straße 1, 28359 Bremen, Germany
[b] BIBA – Bremer Institut für Produktion und Logistik, 28359 Bremen, Germany

ARTICLE INFO

Keywords:
Demand forecasting
Nearest neighbors
Nonlinear dynamics
Inventory management

ABSTRACT

Demand forecasting is of major importance for manufacturing companies since it provides a basis for production planning. However, demand forecasting can be a difficult task because customer demands often fluctuate due to several influences. Methods of nonlinear dynamics have shown promising results in numerous applications but they have mostly been neglected in the context of demand forecasting. This paper evaluates the forecasting performance of local k-nearest neighbor models, which base on the theory of dynamical systems, in a comprehensive empirical study utilizing a large dataset of industrial time series of the M3-Competition. After a broad literature review, the theoretical background is described. Subsequently, different parameter configurations and model selection strategies are compared. A locally constant mean and a locally constant median are compared to locally linear regression models with four different regularization methods and different parameter configurations. In this comparison, the locally constant mean and the locally linear ridge regression with high regularization parameters provide the best trade-offs between forecast accuracy and computation times. Finally, these models achieve a high performance regarding low forecast errors, short computation times as well as high service levels in an inventory simulation compared to established benchmark methods. In particular, they obtain the best results among all applied methods regarding short time series. Moreover, they achieve the lowest errors considering the original accuracy criterion of the M3-Competition. Hence, local k-nearest neighbor models can be regarded as a valid alternative for demand forecasting in an industrial context, accomplishing high forecast accuracy with short computation times.

1. Introduction

Accurate demand forecasts are essential for all companies in a supply chain since several decisions have to be made before the actual values of important variables are known. In addition to forecasts of resource and energy demands (Hahn et al., 2009; Weinert et al., 2011; Hong and Fan, 2016), electricity prices (Aggarwal et al., 2009; Weron, 2014), failure prognosis for maintenance planning (Peng et al., 2010; Sikorska et al., 2011; Fritzsche et al., 2014; Gao et al., 2015) or yield and quality predictions (Lieber et al., 2013; Colledani et al., 2014), forecasts of future customer demands are necessary. In particular, accurate demand forecasts are crucial in build to stock productions, for instance often occurring in the fast moving consumer goods industry. In this context, raw material orders as well as production programs base strongly on demand forecasts and customer demands are directly satisfied from finished goods inventories. Normally, forecasts are calculated on a weekly or monthly basis for a horizon of approximately one year (Fleischmann et al., 2015) based on time series of past customer orders (Tempelmeier, 2008). Due to various influencing factors, demand evolutions are often highly volatile. For example, the demand of a customer depends on the customer's remaining product inventory, the number of available substitute products on the market, the success of marketing campaigns, the current economic, political, social or ecological situation as well as basic agreements (Wiendahl et al., 2007; Porter, 2008). Therefore, forecasting future customer demands can be a difficult task.

A typical demand planning process in a company consists of six steps: (i) data acquisition and preparation, (ii) statistical forecasting, (iii) judgmental forecasting, (iv) consensus forecasting, (v) release of the forecast as well as (vi) evaluation and monitoring (Kilger and Wagner, 2015; Petropoulos et al., 2016). In this context, the step of statistical forecasting is of particular importance since it provides forecasts by automatic algorithms that are used as a basis for the following steps. Therefore, the paper at hand focuses on statistical forecasting and neglects the other steps of demand planning. It has to be noted that this paper considers the forecasting of regular customer demand

* Correspondence to: Hochschulring 20, 28359 Bremen, Germany.
E-mail address: mirko.kueck@uni-bremen.de (M. Kück).

https://doi.org/10.1016/j.ijpe.2020.107837
Received 1 July 2019; Received in revised form 23 May 2020; Accepted 10 June 2020
Available online 12 June 2020

characterized by time series with integer values greater than zero. For applications regarding intermittent demand, which is characterized by a sporadic demand evolution including periods without a demand, such as often observed for spare parts, see (Syntetos and Boylan, 2001; Wallström and Segerstedt, 2010; Kourentzes, 2013, 2014; Van Wingerden et al., 2014).

Frequently, manufacturing companies have to forecast customer demands for thousands of products simultaneously. Hence, automatic forecasting methods have to be applied to calculate big amounts of forecasts. While simple forecasting methods, such as moving averages, simple exponential smoothing or linear regression methods are still often used in industrial practice (Küsters et al., 2006), more sophisticated methods already proved applicability in many practical applications as well as in forecasting competitions (Hyndman, 2020). Autoregressive integrated moving average (ARIMA) models (Hyndman and Khandakar, 2008; Babai et al., 2013; Box et al., 2015; Svetunkov and Boylan, 2019) and complex variants of exponential smoothing (Gardner, 2006; Hyndman et al., 2008; Ferbar Tratar et al., 2016; Kück et al., 2016b; Sbrana and Silvestrini, 2014, 2019) are established methods that have shown potential in several studies, such as the M3-Competition (Makridakis and Hibon, 2000). In addition, machine learning methods, such as artificial neural networks (Crone et al., 2008; Crone and Kourentzes, 2010; Adeodato et al., 2011; Montavon et al., 2012; Kourentzes, 2013; Kourentzes et al., 2014) or support-vector machines (Cortes and Vapnik, 1995; Crone et al., 2008; Lu, 2014) have shown promising results in current studies, such as the NN3-Competition (Crone et al., 2011). However, these methods generally need long computation times and they carry the risk of overfitting when they are applied fully automatically. Moreover, some of these methods need long time series of past customer demands in order to fit their parameters in a suitable way. While demand forecasting can commonly be considered as a big data problem of calculating many forecasts for several products at regular intervals, it is nevertheless often the case that the individual demand histories only comprise a few years. Hence, computationally efficient methods that achieve a stable forecasting performance and do not need long demand histories to fit their parameters suitably would be beneficial.

Despite the variety of methods that have been applied to forecast industrial time series, local k-nearest neighbor models have been mostly neglected so far. Based on a modeling by nonlinear dynamical systems, these methods are able to reconstruct the dynamical properties of a demand evolution as well as influencing factors out of a scalar time series of past customer demands. These so-called methods of nonlinear dynamics have been successfully applied to conduct different tasks in production and logistics systems, such as modeling and control of production systems (Scholz-Reiter et al., 2002; Papakostas et al., 2009) or forecasting sporadic spare parts demand (Nikolopoulos et al., 2016). Furthermore, small-scale studies covering one-step ahead forecasts with small sets of exemplary customer demand time series have already shown promising results of local k-nearest neighbor forecasting models of nonlinear dynamics to forecast regular monthly customer demand (Mulhern and Caprara, 1994; Kück and Scholz-Reiter, 2013; Kück et al., 2014). However, the exemplary results of these studies can only be seen as an indication that these methods could perform well to forecast industrial time series and this assumption has to be proven in a comprehensive study.

The contributions of this paper are the following:

- Proposition of local k-nearest neighbor models for regular monthly demand forecasting: The paper proposes local k-nearest neighbor models as a novel class of methods to forecast regular monthly customer demands of a manufacturing company.
- Parameter comparison and model selection strategies: Two different versions of k-nearest neighbor models are compared, namely locally constant models and locally linear regression models. While standard parameters are used for delay coordinate embedding, different regularization methods with respective regularization parameters are compared regarding their suitability to forecast industrial time series. Moreover, different model selection strategies are proposed.
- Evaluation regarding forecast accuracy: The forecasting performance of two pre-parameterized versions of the local k-nearest neighbor models as well as different individual model selection strategies are evaluated in a comprehensive empirical study utilizing 808 monthly industrial time series of the M3-Competition. As a first criterion of forecasting performance, the methods are evaluated in rolling-origin experiments with forecasts from multiple origins across short, medium and long horizons in comparison to several established benchmark methods. In addition to the evaluations in terms of mean forecast errors, Friedman and Nemenyi tests are conducted to assess significant differences between the performances of the local k-nearest neighbor models and the benchmark methods.
- Evaluation regarding computation times: In contrast to the usual practice in other studies, this paper also evaluates the computation times of the forecasting methods. In this context, mean times per parameter optimization and per forecast are compared.
- Evaluation regarding inventory performance: Another criterion that is often neglected in other studies is the inventory performance achieved based on calculated forecasts. This paper evaluates the local k-nearest neighbor models and the benchmark methods considering achieved service levels and needed safety stocks in an inventory simulation.

The remainder of this paper is structured as follows: Section 2 provides an overview of the existing literature regarding applications of local k-nearest neighbor forecasting models. Section 3 explains the theoretical background. After a parameter comparison and a comparison of different model selection strategies in Section 4, Section 5 comprises a comprehensive empirical study to assess the forecasting performance of the local k-nearest neighbor models in terms of forecast errors and computation times. Section 6 applies the forecasting methods in an inventory simulation and evaluates their performance regarding inventory metrics. Section 7 concludes the paper.

2. Literature overview

Local k-nearest neighbor models belong to the methods of nonlinear dynamics, which originate from the theory of dynamical systems and which base on the paradigm of deterministic chaos. For general overviews regarding methods of nonlinear dynamics, see Kantz and Schreiber (2004), Aguirre and Letellier (2009) and Broer and Takens (2011). Apart from successful applications in modeling, analysis and control of manufacturing systems (Ramadge, 1993; Prabhu and Duffie, 1995; Wiendahl and Scheffczyk, 1999; Scholz-Reiter et al., 2002; Katzorke and Pikovsky, 2006; Cho and Erkoc, 2009; Papakostas et al., 2009) as well as supply chain network applications (Wilding, 1998; Helbing, 2003; Surana et al., 2005; Scholz-Reiter et al., 2006; Donner et al., 2008; Hwarng and Xie, 2008; Klug, 2016; Freitag et al., 2015; Kück et al., 2016a; Chankov et al., 2016), methods of nonlinear dynamics have also been applied in forecasting applications in the form of local k-nearest neighbor models. The k-nearest neighbor ($k$nn) approach was introduced in a technical report by Fix and Hodges (1951) as a classification approach in the context of nonparametric discriminant analysis. However, since the authors did not publish their report at first, the approach only gained popularity after the work of Cover and Hart (1967). Later, the approach was extended from classification to regression and time series forecasting simultaneously in the statistical community (Stone, 1977; Cleveland, 1979; Yakowitz, 1987) as well as in the community of deterministic chaos and dynamical systems (Farmer and Sidorowich, 1987; Casdagli, 1989; Abarbanel

et al., 1990; Sugihara and May, 1990). Afterwards, the approach has been developed further by several authors, for instance to locally linear models based on filtered delay embeddings (Sauer, 1994) or locally linear models with regularization (Kugiumtzis et al., 1998; Kück et al., 2014). Moreover, nearest trajectory strategies have been proposed as an alternative to nearest neighbors (McNames, 1998), approaches for model selection have been developed (Bontempi et al., 1999; McNames, 2002; Jayawardena et al., 2002; Yankov et al., 2006; She and Yang, 2010; Kück and Scholz-Reiter, 2013) and approaches for multiple-step ahead forecasting have been proposed (Bontempi and Ben Taieb, 2011; Ben Taieb et al., 2012). These methods have been used in forecasting applications of economics and finance, natural systems as well as production and logistics. The following subsections review these applications.

2.1. *k*-nearest neighbor forecasting applications regarding economics and finance

Early applications of *k*nn approaches for exchange rate forecasting failed to achieve significant improvements in accuracy compared to standard approaches, such as a random walk model or linear models (Mizrach, 1992; Jaditz and Sayers, 1998). Cecen and Erkal (1996a,b) found that exchange rate forecasts do not show a deterministic nonlinear dependence but rather a nonlinear stochastic behavior, which may be a reason for these unsatisfactory results. In contrast to the early approaches, more recent studies were able to show more potential of *k*nn approaches for economics and finance applications. Fernández-Rodríguez et al. (1999) applied a local *k*nn approach for exchange rate forecasting, achieving significantly better predictions than a random walk and marginally better predictions than an ARIMA model. Lisi and Medio (1997) outperformed a random walk by combining a singular spectrum analysis with a locally linear method to forecast exchange rate series of the major currencies. Cao and Soofi (1999) applied delay coordinate embedding and a locally linear *k*nn regression to forecast dollar exchange rate returns. Some forecasts outperformed a mean value predictor, however not statistically significant. Meade (2002) compared three *k*nn approaches, a locally weighted linear regression and a linear generalized autoregressive conditional heteroscedasticity (GARCH) model to a random walk for short-term exchange rate forecasts using data frequencies from daily to half-hourly. The different methods achieved significantly better results than the random walk. However, no evidence was found that the exchange rate behavior would be better represented by a nonlinear model than by a linear model. Lisi and Schiavo (1999) compared an artificial neural network and a locally linear *k*nn regression in an application to forecast the monthly exchange rates of the four major European currencies from 1973 to 1995. The methods obtained forecasts that were statistically equivalent in most cases and significantly better than those of a random walk. Similarly, Alvarez-Diaz (2008) applied different local *k*nn methods as well as global methods, such as neural networks and genetic programming, to predict the dynamic evolution of the Yen/US$ and Pound Sterling/US$ exchange rates. The results revealed that the methods were able to predict one-period ahead but not more periods ahead. Moreover, no important predictive differences between local and global methods were found.

2.2. *k*-nearest neighbor forecasting applications regarding natural systems

In a collection of competitions at the Santa Fe Institute, in total, six time series had to be forecast, including a laser dataset, a physiological dataset from a patient with sleep apnea, a high-frequency exchange rate dataset and an astrophysical dataset from a variable star (Weigend and Gershenfeld, 1992). The time series were very long, comprising between 1000 and 34,000 points. Nearest neighbor approaches, such as the locally linear regression model by Sauer (1994) performed well in the competitions. Jursa and Rohrig (2008) applied a neural network and a nearest neighbor model to forecast the power output of a wind farm. They found that combining the forecasts of both models led to better forecast accuracy than the individual forecasts. Ragwitz and Kantz (2002) showed that under certain conditions, a scalar time series obtained from a vector-valued Markov process can be modeled as a finite-memory Markov process in the observable. Moreover, they showed that the state transition rules can be computed by a locally constant *k*nn model. Kantz et al. (2004) showed that this method can be used to forecast turbulent gusts of wind. Dimri et al. (2008) applied a nearest neighbor method to forecast the probability of precipitation and its quantity over the western Himalayas. In an empirical study, the probability of occurrence of precipitation was predicted well by the method. Singh et al. (2015) optimized the parameters of a nearest neighbor model by an artificial bee colony algorithm and applied the model for avalanche forecasting regarding two climatologically diverse avalanche prone regions of the Indian Himalaya. The method obtained significant gains in forecast accuracy in terms of the Heidke skill score.

2.3. *k*-nearest neighbor forecasting applications regarding production and logistics

Local *k*nn approaches have been used in several different applications in the context of production and logistics, such as electricity price forecasting, traffic prediction, TV audience shares, cash withdrawals as well as demand forecasting. Lora et al. (2007) presented a weighted nearest neighbor model to forecast hourly electricity market prices of the next day. In an evaluation regarding the Spanish electricity market during 2002, the model achieved promising forecasts compared to several benchmark methods, such as a neural network, a Neuro-Fuzzy system, a GARCH model and an ARIMA model with and without wavelet transformation. Al-Qahtani and Crone (2013) proposed a multivariate regression approach to forecast UK electricity demand by combining a locally constant *k*nn model with a coding of working versus non-working and weekend days as binary dummy variables. The approach outperformed different random walk and moving average models as well as a univariate *k*nn model in forecasting the next 24-hour electricity load. Sun et al. (2003) applied local modeling approaches for short-term traffic prediction. In their study based on 32-day traffic-speed data collected at five minute intervals, they found that a locally linear regression model achieved a better forecasting performance than locally constant and kernel smoothing methods. Nikolopoulos et al. (2007) compared different methods to forecast the impact on Greek TV audience shares of programs showing sport events. In this study, a simple nearest neighbor model outperformed more complex and computationally demanding methods, such as a multiple linear regression model and neural networks. The study indicates that *k*nn models provide forecasts of high quality automatically, while more complex methods demand high expertise for suitable parameter configuration. Wichard (2011) combined an ensemble of nearest trajectory models, a neural network ensemble and a model for the 7-day cycle in a hybrid model ensemble to forecast daily cash money withdrawals at cash-machines in the NN5 competition. The approach ranked sixth out of 19 competitors among the computational intelligence methods and ninth among 27 competitors overall.

Nikolopoulos et al. (2016) applied locally constant *k*nn models to forecast intermittent demand. Based on a large dataset from automotive industry as well as a simulated dataset, they concluded that the *k*nn models should be applied in cases with repetitive patterns but standard approaches for sporadic time series forecasting should be preferred if no patterns are present. Mulhern and Caprara (1994) applied a multivariate nearest neighbor model to forecast weekly sales of a single brand of cake mix in a grocery store. In a rolling-origin evaluation with one-step ahead forecasts of a time series spanning a period of nearly two years, the nearest neighbor approach outperformed an ARIMA model since it was able to consider time effects. Kück and Scholz-Reiter (2013) developed a genetic algorithm to optimize the parameters of

local $k$nn models. In a small-scale study based on four time series of the M3-Competition and four synthetic time series, locally constant and locally linear $k$nn models achieved better one-step ahead forecasts with optimized parameters than with rule-of-thumb parameters. In a subsequent study, Kück et al. (2014) showed that combining the locally linear models with regularization methods leads to an increased robustness against noise in the data. The empirical investigations again based on eight time series and one-step ahead forecasts. In addition to these small-scale studies, Scholz-Reiter et al. (2014) developed a meta-learning approach to select an individual forecasting method out of six candidates based on time series characteristics. The method candidates were a locally constant model, a locally linear model, an ARIMA model, a neural network, an exponential smoothing method and a random walk. In a study based on a large-scale dataset of the M3-Competition, the meta-learning approach achieved a better forecasting performance than the six individual forecasting methods as well as a conventional model selection method. However, the study again only considered one-step ahead forecasts.

2.4. Summary and research gap

The literature overview shows that local $k$nn forecasting models have been successfully applied in applications of economics and finance, natural systems as well as production and logistics. While early approaches sometimes showed mediocre performance, more recent local $k$nn approaches often achieved good forecasting results. Albeit, in the context of customer demand forecasting, few studies have applied local $k$nn models so far. Apart from a study to forecast sporadic spare parts demands, there have been some attempts to assess the performance of local $k$nn models to forecast regular customer demands. However, these studies were based on small datasets and they were limited to one-step ahead forecasting. Moreover, the forecasting performance was only evaluated in terms of conventional forecast error measures. Considerations concerning the statistical significance of the results as well as comparisons in terms of computation times and inventory simulations have been neglected so far. The paper at hand considers these research gaps regarding the applicability of local $k$nn models in customer demand forecasting applications.

3. Theoretical background

This section describes the necessary steps to build a local k-nearest neighbor ($k$nn) forecasting model. After an introductory overview, the steps of delay coordinate embedding, $k$nn searching and building a local $k$nn forecasting model are detailed.

3.1. Overview

Fig. 1 illustrates the process of forecasting a future value $\hat{y}_{T+h}$ of a given time series $\mathbf{y} = \{y_1, \dots, y_T\}$ by a local $k$nn forecasting model. The process consists of three main steps: (i) delay coordinate embedding, (ii) $k$nn searching, and (iii) building a local $k$nn model, which is then applied to forecast a future time series value.

Delay coordinate embedding constructs a so-called trajectory matrix $\mathbf{Y}_{d,m}$ containing rows called delay coordinate vectors, which are time-lagged segments of the original time series. The length m of these rows is called embedding dimension and the parameter d is called delay time. For instance, $d = 6$ means that a row of the trajectory matrix contains every sixth value of the original time series. For a detailed description of delay coordinate embedding, see Section 3.2. For convenience, Fig. 1 bases on delay coordinate vectors with delay time $d = 1$ and length $m = 8$.

In the second step, the nearest neighbors to the query vector are computed. The query vector is colored in green and the nearest neighbors are cross-hatched in green. The query vector is the last delay coordinate vector of the trajectory matrix and it ends with the last value y_T of the original time series. The nearest neighbors are the delay coordinate vectors that are similar to the query vector, i.e. their Euclidean distances to the query vector are small. In Fig. 1, $k = 3$ nearest neighbors are shown. The process of $k$nn searching is described in Section 3.3.

The third step is building a local $k$nn forecasting model based on the k nearest neighbors and their future evolutions, which are cross-hatched in gray. Fig. 1 shows the application of a locally constant $k$nn model. This means that forecasts are calculated as the mean values of the neighbors' futures. In addition to locally constant models, this paper also applies locally linear regression models, which compute the forecasts after building a linear regression model of the nearest neighbors and their futures. For the reason of better visibility, Fig. 1 shows the direct forecasts of horizons $h = 1, \dots, 4$ based on one delay coordinate embedding and one $k$nn search. However, this paper does not consider direct but iterative local $k$nn forecasting approaches, which conduct the three steps of delay coordinate embedding, $k$nn searching and building a forecasting model for each forecast value individually. For a detailed description of local $k$nn forecasting models, see Section 3.4.

Fig. 2 shows a flow chart illustrating the steps that have to be taken in order to forecast future time series values by a local $k$nn model. The process starts with a parameter initialization, which is shown in Fig. 3. The time series of past customer orders $\mathbf{y} = \{y_1, \dots, y_T\}$ is given. A model *order* for the forecasting model has to be specified either as 0 for a locally constant or as 1 for a locally linear $k$nn forecasting model. The parameters *regtype* and *regpar* are only considered in case of a locally linear model ($order = 1$). *regtype* specifies the regularization type and can be one of the following: 'none' (ordinary least squares regression), 'PCR' (principal component regression), 'PLSR' (partial least squares regression), 'RR (ridge regression) or 'LASSO' (least absolute shrinkage and selection operator method). *regpar* defines the respective regularization parameter M in case of PCR and PLSR or λ in case of RR and LASSO. *regpar* will be ignored if *regtype* is 'none'. H defines the maximum forecast horizon and h is the index of the current forecast horizon in the loop of Fig. 2. Since industrial data with a regular demand are considered in this paper, the parameter *rounded* is fixed to 1, which means that each forecast value is rounded to an integer. For the same reason, the parameter *limits*, which defines the limits for a valid forecast value, is fixed to the interval $[1, \infty)$. If a forecast value is out of these limits, an alternative forecast function is applied, which is specified by *alternativefunction*. In this paper, a random walk forecast is performed as alternative forecast, which means that the last value of the time series is used as a forecast of the following future value. In addition to the initialized parameters, appropriate values for the delay time d, the embedding dimension m and the number of nearest neighbors k have to be found. For this purpose, the following standard strategies to compute reasonable values for these parameters have been chosen (Kantz and Schreiber, 2004): The delay time d is computed as the first minimum of the average mutual information of the time series $\mathbf{y}$ (Fraser and Swinney, 1986). The embedding dimension m is calculated by the false nearest neighbors algorithm (Kennel et al., 1992) and $k = m + 1$.

Fig. 2 shows that after the parameter initialization, a loop is executed to compute the forecast values $\hat{y}_{T+h}$ iteratively for $h = 1, \dots, H$. In each iteration, three main steps are performed. Firstly, a trajectory matrix is computed by delay coordinate embedding, which is described in Section 3.2. Secondly, the distances of all row vectors to the last row vector in the trajectory matrix, i.e. the query vector, are calculated and the k nearest neighbors of the query vector are determined. Section 3.3 details this process. Thirdly, a local $k$nn forecasting model is build and applied to forecast the future value $\hat{y}_{T+h}$, which is described in Section 3.4. Afterwards, it has to be checked if the calculated forecast value is valid. It is rounded to an integer and if it is not inside the predefined limits of $[1, \infty)$, the alternative forecast function (random walk) is applied to forecast a valid forecast value. If $h < H$, the forecast value $\hat{y}_{T+h}$ is appended to the time series $\mathbf{y}$, h is increased by 1 and the next iteration starts. If $h = H$, the iteration is finished and the complete process ends with the output of the forecast values $\hat{y}_{t+1}, \dots, \hat{y}_{t+H}$.

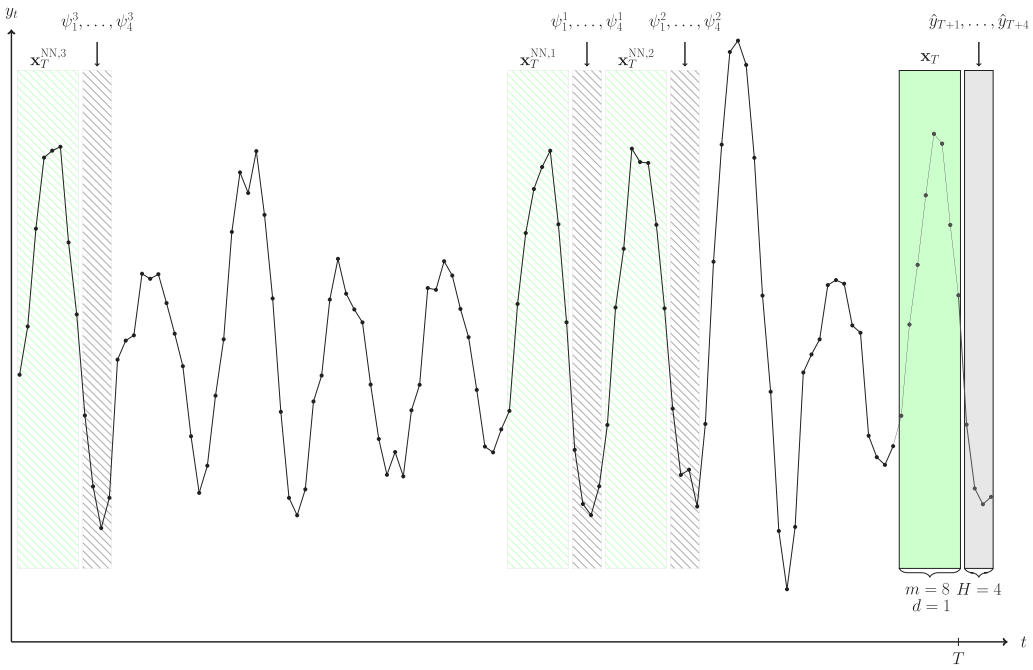


Fig. 1. Illustration of local k-nearest neighbor forecasting. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Delay coordinate embedding

The first step of building a local $k$nn forecasting model is delay coordinate embedding, which bases on the theory of nonlinear dynamics and is also called phase space reconstruction (Kantz and Schreiber, 2004). A dynamical system considers various components that have an impact on its evolution. To describe this evolution in detail, all dependencies have to be known, which usually cannot be assured. Frequently, only measurements of one component of the dynamical system are known. In this paper, only the availability of a successive equidistant scalar time series of past customer orders

$$\mathbf{y} = \{y_1, y_2, \ldots, y_T\} \tag{1}$$

until the present time point T is assumed. The demand evolution embodies one component of the dynamical system, which describes the evolution of a whole production and delivery system. However, under quite general assumptions, the embedding theorem of Takens (1981) and its generalization (Sauer et al., 1991) state that, under the supposition of a deterministic dynamical system, the whole system's evolution can be reconstructed out of a scalar time series of one component of the system. A vector

$$\mathbf{x}_t = \mathbf{x}_{t,m,d} = [x_{t,1}, \ldots, x_{t,m}] = [y_{t-(m-1)d}, y_{t-(m-2)d}, \ldots, y_{t-d}, y_t] \in \mathbb{R}^{1\times m} \tag{2}$$

is called delay coordinate vector of length m corresponding to time point t. The parameter m of the delay coordinate vector is called embedding dimension and d is called delay time. A delay coordinate vector $\mathbf{x}_t$ is a segment of the original time series $\mathbf{y}$. While the original time series $\mathbf{y}$ involves T measurements , the delay coordinate vector contains m of the T components with successive time distance d. The embedding dimension m has to be large enough to achieve a sufficient unfolding of the attractor of the dynamical system but not too large in order to avoid redundancies that could confuse local forecasting models. Similarly, on the one hand, the delay time d should be large enough to prevent too strong correlations between successive elements of delay coordinate vectors. On the other hand, it should not be too large to avoid a complete independence of successive elements. In this paper, m is calculated by the false nearest neighbors algorithm (Kennel et al., 1992) and d is obtained as the first minimum of the average mutual information (Fraser and Swinney, 1986). Further methods to choose these parameters can be found in (Kantz and Schreiber, 2004).

The embedding theorems justify a smooth functional dependence

$$y_{t+1} = \tau_1(\mathbf{x}_t) = \tau_1(y_{t-(m-1)d}, y_{t-(m-2)d}, \ldots, y_{t-d}, y_t) \tag{3}$$

assumed that the time series $\mathbf{y}$ is generated by a smooth deterministic dynamical system and m as well as d are chosen appropriately (Takens, 2010; Kantz and Schreiber, 2004). Hence, a forecasting model for future values of the given time series can be built based on the delay coordinate embedding. For this purpose, all delay coordinate vectors of length m with successive distance d are built out of the time series $\mathbf{y}$ and stored in the so-called trajectory matrix

$$\mathbf{Y}_{d,m} = \begin{pmatrix} y_1 & y_{1+d} & \cdots & y_{1+(m-2)d} & y_{1+(m-1)d} \\ y_2 & y_{2+d} & \cdots & y_{2+(m-2)d} & y_{2+(m-1)d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{t-(m-1)d} & y_{t-(m-2)d} & \cdots & y_{t-d} & y_t \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_t & y_{t+d} & \cdots & y_{t+(m-2)d} & y_{t+(m-1)d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{T-(m-1)d} & y_{T-(m-2)d} & \cdots & y_{T-d} & y_T \end{pmatrix}. \tag{4}$$

The query vector

$$\mathbf{x}_T = [x_{T,1}, \ldots, x_{T,m}] = [y_{T-(m-1)d}, y_{T-(m-2)d}, \ldots, y_{T-d}, y_T] \tag{5}$$

is the delay coordinate vector corresponding to the present state. After computing the trajectory matrix, the dynamical properties of a given time series are reconstructed and therefore, the first step of building a local $k$nn forecasting model is accomplished.

3.3. k-nearest neighbor searching and weights

The second step of the described forecasting approach is searching for the k nearest neighbors of the query vector $\mathbf{x}_T$, which is the last vector of the trajectory matrix $\mathbf{Y}_{d,m}$. The other $T-(m-1)d-1$ vectors of the trajectory matrix are sorted by their distance to the query vector $\mathbf{x}_T$. In this context, different distance metrics, such as the Euclidean distance, the diagonally weighted Euclidean distance or any p-norm,

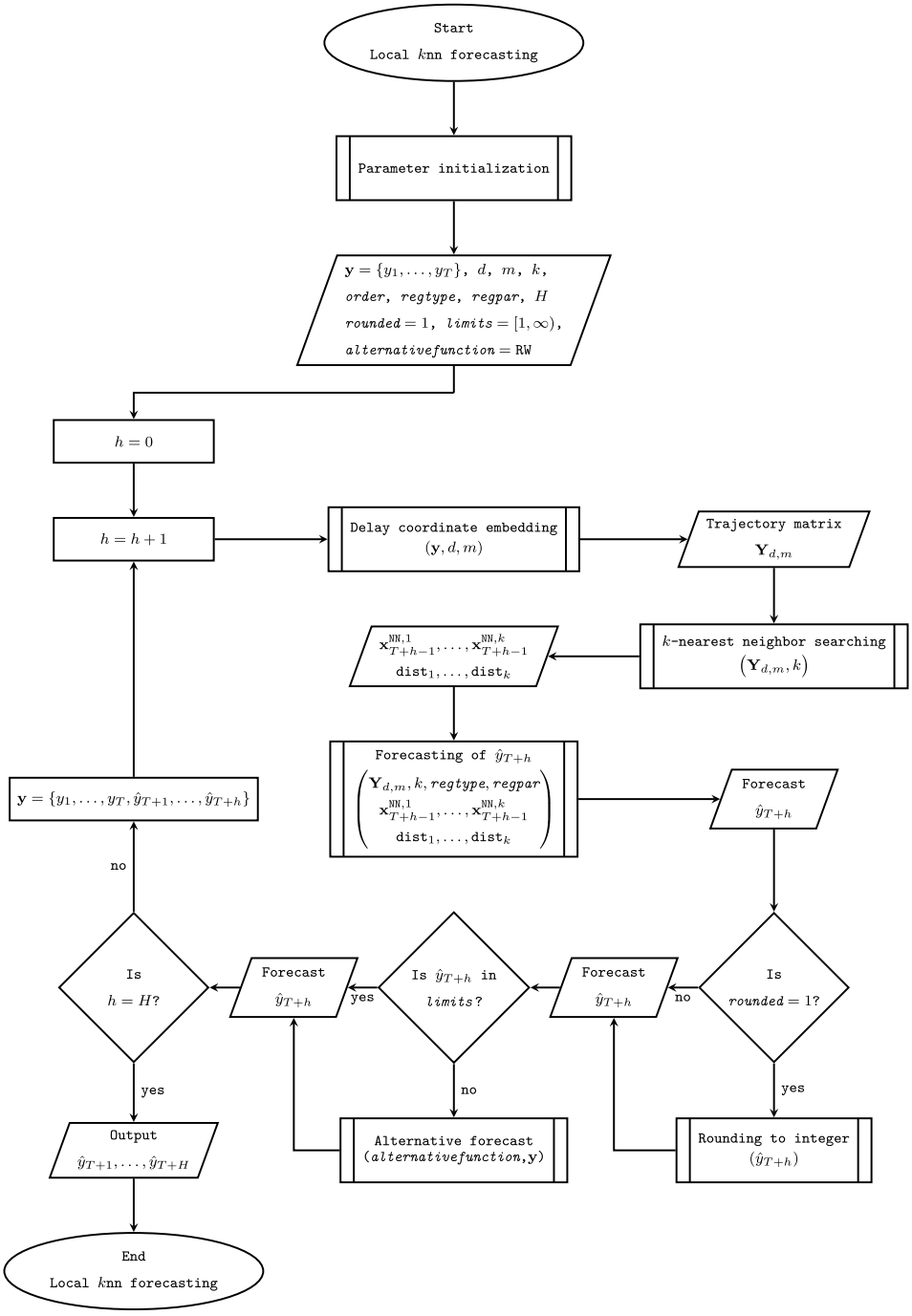


Fig. 2. Flow chart of the necessary steps to forecast future values of a given time series by a local k-nearest neighbor forecasting model.

could be used (McNames, 1999; Engster, 2011). In this paper, the Euclidean distance

$$\text{dist}(\mathbf{x}_T, \mathbf{x}_i) = \sum_{j=1}^{m} \left(y_{T-(m-j)d} - y_{i-(m-j)d} \right)^2 \tag{6}$$

is used. Now, the k nearest neighbors $\mathbf{x}_T^{\text{NN},1}, \ldots, \mathbf{x}_T^{\text{NN},k}$ of $\mathbf{x}_T$ are determined, where the ith nearest neighbor is the delay coordinate vector that has the ith shortest distance dist_i to the query vector $\mathbf{x}_T$.

In addition to the embedding parameters m and d, the number k of nearest neighbors that have to be considered has to be selected suitably. In general, a small value of k leads to a low bias of a forecasting model but also to a high variance. Vice versa, a high value of k leads to a low variance but a high bias. In this paper, the parameter is calculated as $k = m + 1$ (Kantz and Schreiber, 2004).

On the one hand, the k nearest neighbors can be used directly to build forecasting models. On the other hand, the computed distances of the neighbors to the query vector offer the possibility to weight the importance of the neighbors for the forecasting model according to the distances. In this paper, equal weights for all neighbors

$$\mathbf{w}_{\text{equal}} = (1, \ldots, 1) \in \mathbb{R}^k \tag{7}$$

are used. Another common approach is weighting the importance of the neighbors according to the bisquare weight function. However, this approach performed worse than using equal weights in a preliminary experiment although being more complicated. Therefore, this paper uses equal weights for the nearest neighbors.

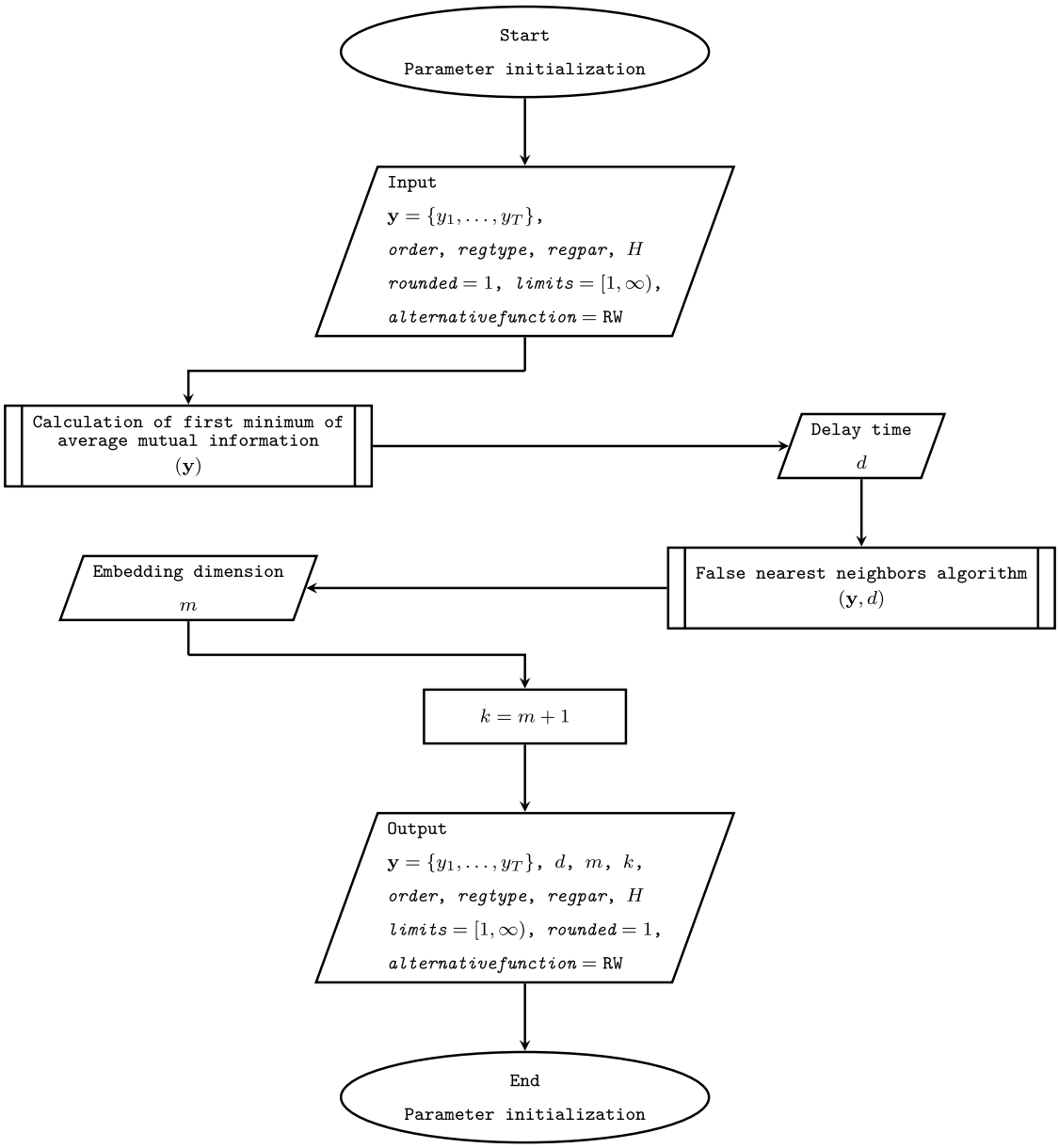


Fig. 3. Flow chart of parameter initialization for local k-nearest neighbor forecasting.

3.4. Local k-nearest neighbor forecasting models

After delay coordinate embedding and $k$nn searching, a local $k$nn forecasting model can be built. By using the trajectory matrix $\mathbf{Y}_{d,m}$, the k nearest neighbors can be mapped h steps into the future in the reconstructed phase space to obtain the values $\tau_h\left(\mathbf{x}_T^{\mathrm{NN},1}\right), \ldots, \tau_h\left(\mathbf{x}_T^{\mathrm{NN},k}\right)$. Suppose the ith nearest neighbor is the vector

$$\mathbf{x}_T^{\mathrm{NN},i} = \mathbf{x}_j = \left[y_{j-(m-1)d}, \ldots, y_{j-d}, y_j\right]. \tag{8}$$

Then a mapping of h steps into the future means to consider

$$\tau_h\left(\mathbf{x}_T^{\mathrm{NN},i}\right) = y_{j+h}. \tag{9}$$

The vector that contains the hth future values of the k nearest neighbors is denoted as

$$\boldsymbol{\psi}_h = \left(\psi_h^1, \ldots, \psi_h^k\right)^\top = \left(\tau_h\left(\mathbf{x}_T^{\mathrm{NN},1}\right), \ldots, \tau_h\left(\mathbf{x}_T^{\mathrm{NN},k}\right)\right)^\top. \tag{10}$$

By using the future evolutions (10) of the k nearest neighbors, different forecasting models can be built that approximate the global nonlinear evolution locally (Farmer and Sidorowich, 1987; Sauer, 1994; Kück and Scholz-Reiter, 2013; Kück et al., 2014). In this paper, locally constant forecasting models and locally linear regression models are applied.

3.4.1. Locally constant k-nearest neighbor models

The simplest approach to build a local $k$nn forecasting model is to use a locally constant model, which is also called a local averaging model (McNames, 2002). In this context, the most common approach is computing the sample mean of the neighbors' futures

$$\hat{y}_{T+h}^{\mathrm{LC,mean}} = \frac{1}{k}\sum_{i=1}^{k} \tau_h\left(\mathbf{x}_T^{\mathrm{NN},i}\right) = \frac{1}{k}\sum_{i=1}^{k} \psi_h^i. \tag{11}$$

In addition, this paper introduces an alternative approach that computes the median of the future evolutions of the nearest neighbors. For this purpose, the vector $\boldsymbol{\psi}_h$ of the neighbors' futures has to be sorted to obtain

$$\boldsymbol{\psi}_h' = \left(\psi_h^{(1)}, \psi_h^{(2)}, \ldots, \psi_h^{(k)}\right)^\top, \text{ with } \psi_h^{(i)} \leq \psi_h^{(i+1)} \text{ for } i \in \{1, \ldots, k-1\}. \tag{12}$$

The locally constant median is computed as

$$\hat{y}_{T+h}^{\text{LC,median}} = \begin{cases} \psi_h^{\left(\frac{k+1}{2}\right)}, & \text{if } k \text{ is odd} \\ \frac{1}{2}\left(\psi_h^{\left(\frac{k}{2}\right)} + \psi_h^{\left(\frac{k}{2}\right)+1}\right), & \text{if } k \text{ is even.} \end{cases} \tag{13}$$

3.4.2. Locally linear k-nearest neighbor regression

In contrast to locally constant $k$nn models, locally linear $k$nn regression models build a linear model

$$\boldsymbol{\psi}_h = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\psi}_h = \begin{pmatrix} \psi_h^1 \\ \vdots \\ \psi_h^k \end{pmatrix} = \begin{pmatrix} \tau_h\left(\mathbf{x}_T^{\text{NN},1}\right) \\ \vdots \\ \tau_h\left(\mathbf{x}_T^{\text{NN},k}\right) \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} 1 & \mathbf{x}_T^{\text{NN},1} \\ \vdots & \vdots \\ 1 & \mathbf{x}_T^{\text{NN},k} \end{pmatrix} \tag{14}$$

of the neighbors' futures $\boldsymbol{\psi}_h \in \mathbb{R}^k$ on the neighbors' components $\mathbf{X} \in \mathbb{R}^{k\times(m+1)}$, where $\boldsymbol{\beta} = \left(\beta_0, \beta_1, \ldots, \beta_m\right)^\top \in \mathbb{R}^{m+1}$ denotes a coefficient vector and $\boldsymbol{\epsilon} \in \mathbb{R}^k$ is a white noise process. After calculating a suitable coefficient vector $\hat{\boldsymbol{\beta}}$, a h-step-ahead prediction is computed by

$$\hat{y}_{T+h} = \hat{\beta}_0 + \sum_{j=1}^{m} x_{T,j}\hat{\beta}_j = \left(\mathbf{1}\ \mathbf{x}_T\right)\hat{\boldsymbol{\beta}}, \tag{15}$$

where, $x_{T,j}$ denotes the jth component of the query vector $\mathbf{x}_T$ and $\mathbf{1} = (1, \ldots, 1)^\top \in \mathbb{R}^{m\times 1}$. According to the Gauss–Markov theorem (Hastie et al., 2009), the best linear unbiased estimator for the coefficient vector $\boldsymbol{\beta}$ is the one that minimizes the residual sum of squares

$$\text{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{k}\left(\tau_h\left(\mathbf{x}_T^{\text{NN},i}\right) - \beta_0 - \sum_{j=1}^{m} x_{T,j}^{\text{NN},i}\beta_j\right)^2, \tag{16}$$

where $x_{T,j}^{\text{NN},i}$ denotes the jth component of the ith nearest neighbor $\mathbf{x}_T^{\text{NN},i}$. This so-called ordinary least squares (OLS) coefficient vector can be computed as

$$\hat{\boldsymbol{\beta}}^{\text{OLS}} = \underset{\boldsymbol{\beta}}{\text{argmin}}\left(\text{RSS}(\boldsymbol{\beta})\right) = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\boldsymbol{\psi}_h. \tag{17}$$

In many cases of observed real world time series data, columns of the matrix $\mathbf{X}$ are collinear, which implies an ill-conditioned matrix inverse of $\mathbf{X}^\top\mathbf{X}$ and thus some regression coefficients as well as the model output can be very large (McNames, 1999). In order to solve this problem and make the locally linear model more robust against noise, regularization methods can be applied (Kugiumtzis et al., 1998; Kück et al., 2014). These methods bias the estimator of the regression coefficients to reduce the variance as well as the mean squared error of the estimator and to prevent the model from overfitting. They can be divided into methods using derived input directions and shrinkage methods (Hastie et al., 2009). Principal component regression (PCR) and partial least squares regression (PLSR) use derived input directions and base the regression on a small number of linear combinations of the original inputs (Hastie et al., 2009; Kugiumtzis et al., 1998). Shrinkage methods, such as ridge regression (RR) and the least absolute shrinkage and selection operator method (LASSO), shrink the regression coefficients by imposing a penalty on their size (Tibshirani, 1996; Hastie et al., 2009).

4. Parameter comparison and model selection strategies

This section determines valuable parameter configurations and model selection strategies for the local $k$nn models to forecast monthly customer demands. The section starts with an experiment description, considering the compared parameter configurations, the used dataset and the performance criteria, which are mean forecast accuracies and computation times. Afterwards, the results of the parameter comparison are shown and suitable model selection strategies are determined.

4.1. Experiment description

4.1.1. Forecasting methods and parameter ranges

Section 4.2 compares different locally constant and locally linear $k$nn forecasting models in order to determine parameter configurations that achieve a high performance, i.e. high forecast accuracy at different horizons as well as short computation times. As locally constant (LC) models, the mean and the median are compared. As locally linear (LL) models, OLS as well as the regularization methods PCR, PLSR, RR and LASSO are compared using different regularization parameters. The regularization parameter of PCR and PLSR specifies the minimum percentage of the cumulative variance of all principal components or partial least square components that the used components shall cover. The regularization parameter of RR and LASSO specifies the amount of shrinkage. Table 1 shows the different parameter configurations that are compared in Section 4.2. These parameters provide a diverse cross section of two different locally constant models as well as locally linear regression models with low, medium and high regularization. For the determination of model selection strategies in Section 4.3, different sets of possible model candidates are used. The parameter configurations in Table 1 that are marked with a * are the model candidates that are used by the model selection strategies in the empirical studies in Sections 5 and 6. For more information about the model selection strategies, see Section 4.3.

4.1.2. Dataset

The empirical dataset used to evaluate the performance of different parameter configurations of the local $k$nn forecasting models is a subset of the M3-Competition dataset (Makridakis and Hibon, 2000). The complete M3 dataset consists of 3003 univariate time series, which were recorded at different frequencies (yearly, quarterly, monthly, other) from different fields of application (micro, industry, macro, finance, demographic, other). In the competition, forecasting experts received the time series and had to forecast future values of each series, which were known by the organizers but not by the participants. The real future values were used to evaluate the accuracy of the submitted forecasts and they were disclosed after the end of the competition.

In this paper, the subset of 808 monthly time series with the categories industry and micro is used. This subset mainly consists of real industrial data of demands, shipments, production and sales. In general, the structure of supply chains allows to translate non-demand data into the demand for some other products, such as raw material or spare parts, necessary to produce new products. Therefore, it can be assumed that all of the 808 utilized time series closely resemble customer demands for a product. It has to be noted that some of the time series represent services, such as maintenance activities or deliveries, instead of physical products. However, these time series are also used for the evaluations of this paper in order to achieve comparable results to earlier and future studies using the same dataset. Just as in the competition, each of the 808 time series is divided into a training set

$$\mathbf{y}_{\text{train}} = \left\{y_1, y_2, \ldots, y_T\right\} \tag{18}$$

and a test set

$$\mathbf{y}_{\text{test}} = \left\{y_{T+1}, y_{T+2}, \ldots, y_{T+H}\right\}. \tag{19}$$

The training sets of the 808 time series consist of 50 to 126 values and each test set consists of $H = 18$ values. A time series is considered as long, if the training set comprises more than 100 values and as short otherwise. This leads to a separation of the complete set into subsets of 278 short and 530 long time series.

Table 1
Compared model configurations of the different parameter sets of the local k-nearest neighbor forecasting models.

Parameter set	Model configurations
Locally constant	Mean*, Median*
Locally linear OLS	–
Locally linear PCR	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,0.75, 0.8, 0.85, 0.9, 0.95
Locally linear PLSR	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,0.75, 0.8, 0.85, 0.9, 0.95
Locally linear RR	0.01, 0.05, 0.1, 0.5, 1, 5*, 10*, 50*, 100*, 500, 1000, 5000
Locally linear LASSO	0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000, 5000

*The model candidates that are used for the final model selection strategies.

4.1.3. Performance criteria

In order to determine suitable parameter configurations for the local $k$nn forecasting models, two performance criteria are considered in this section, namely mean forecast accuracy and mean computation time per forecast. The forecast accuracy is measured as the average of the rolling-origin symmetric mean absolute percentage error of all forecasts of horizons $h = h_{\min}, \dots, h_{\max}$ from different origins $o = o_{\min}, \dots, o_{\max}$ on the test set. For this purpose, the symmetric mean absolute percentage error (sMAPE) calculated for the actuals $\mathbf{y}$ and the forecasts $\hat{\mathbf{y}}$ of horizons $h = h_{\min}, \dots, h_{\max}$ from origin o is defined as

$$\begin{aligned} \text{sMAPE}_o^{h_{\min},h_{\max}}(\mathbf{y},\hat{\mathbf{y}}) &= \frac{1}{H}\sum_{h=h_{\min}}^{h_{\max}} \frac{200|y_{o+h}-\hat{y}_{o+h}|}{|y_{o+h}|+|\hat{y}_{o+h}|} \\ &= \frac{1}{H}\sum_{t=o+h_{\min}}^{o+h_{\max}} \frac{200|y_t-\hat{y}_t|}{|y_t|+|\hat{y}_t|}, \end{aligned} \tag{20}$$

where $H = h_{\max} - h_{\min} + 1$. This is a comparatively unbiased, scale-independent error measure, which allows comparing accuracy across multiple time series of different levels (Hyndman and Athanasopoulos, 2018). The sMAPE is defined for the forecasts from one fixed forecasting origin o. However, forecasts calculated from a single origin can be corrupted by occurrences unique to this origin. Hence, a rolling-origin evaluation is applied (Tashman, 2000). For this purpose, all sMAPEs across horizons $h = h_{\min}, \dots, h_{\max}$ are calculated from multiple forecasting origins $o = o_{\min}, \dots, o_{\max}$ and all errors are averaged to form the rolling-origin sMAPE (RO-sMAPE)

$$\text{RO-sMAPE}_{o_{\min},o_{\max}}^{h_{\min},h_{\max}}(\mathbf{y},\hat{\mathbf{y}}) = \frac{1}{O}\sum_{o=o_{\min}}^{o_{\max}} \text{sMAPE}_o^{h_{\min},h_{\max}}(\mathbf{y},\hat{\mathbf{y}}), \tag{21}$$

with $O = o_{\max} - o_{\min} + 1$. After computing the RO-sMAPE for a set of actual time series $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_P\}$ and a set of forecasts $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_P\}$, the mean RO-sMAPE (Mean-RO-sMAPE) is used as a valid criterion for measuring the forecast accuracy across multiple horizons from different origins and for different time series:

$$\text{Mean-RO-sMAPE}_{o_{\min},o_{\max}}^{h_{\min},h_{\max}}\left(\mathbf{Y},\hat{\mathbf{Y}}\right) = \frac{1}{P}\sum_{i=1}^{P} \text{RO-sMAPE}_{o_{\min},o_{\max}}^{h_{\min},h_{\max}}\left(\mathbf{y}_i,\hat{\mathbf{y}}_i\right). \tag{22}$$

In this section, the Mean-RO-sMAPE for $h_{\min} = 1$, $h_{\max} = H = 12$, $o_{\min} = T$ and $o_{\max} = T + 6$ is applied:

$$\text{Mean-RO-sMAPE}_{T,T+6}^{1,12}\left(\mathbf{Y},\hat{\mathbf{Y}}\right) = \frac{1}{808}\sum_{i=1}^{808}\frac{1}{7}\sum_{o=T}^{T+6} \text{sMAPE}_o^{1,12}\left(\mathbf{y}_i,\hat{\mathbf{y}}_i\right). \tag{23}$$

This means that for each test set of 18 points, 12-step-ahead forecasts are calculated from 7 different origins leading to 84 forecasts per time series, whose average is calculated by the Mean-RO-sMAPE. This measure represents the forecast accuracy of a model for a wide range of horizons.

In addition to high forecast accuracy, it is beneficial if a specific parameter configuration of a local $k$nn forecasting model needs a short computation time to calculate a forecast. Therefore, the mean time per forecast of a specific parameter configuration for all computed forecasts for all 808 time series is applied as the second performance criterion. For the model selection strategies in Section 4.3, the times per parameter optimization are considered in addition to the times per forecast. A parameter optimization of a local $k$nn model consists of determining the model parameters d, m and k. For the model selection strategies, the training set of a time series is divided into a smaller training set and a validation set and the Mean-RO-sMAPEs of different models are compared on the validation set. In this case, the time per optimization consists of the times per optimization of all compared model candidates, the time to forecast all values of the validation set with all model candidates, the time to compare the RO-sMAPEs of the model candidates and the time per optimization of the model candidate that is selected to forecast the test set of the time series. A parameter optimization has to be conducted once for a time series and then the forecasting method is applied with the resulting parameter set to compute the forecasts for different horizons from different origins. In contrast, a time per forecast measures the computation time needed to forecast one value. Therefore, for instance, this computation time is needed 18 times for a fixed origin evaluation with $h_{\min} = 1$ and $h_{\max} = 18$ or 84 times for a rolling-origin evaluation with $h_{\min} = 1$, $h_{\max} = 12$ and $o_{\max} = T + 6$.

All experiments of this paper have been conducted using a standard Windows computer with Intel(R) Core(TM) i5-4300U CPU @1.90 GHz and 8 GB RAM. The local $k$nn models have been implemented prototypically in Matlab. For a professional application in an industrial context, an implementation in a higher programming language would offer potential for reduced computation times. The computation time for a single time series is not deterministic since it depends on all services currently running on the computer and therefore it will change slightly if the same experiment is conducted again. However, by averaging the mean computation times for all forecasts, all parameter optimizations and all time series, it can be assumed that the average computation times are a robust measure of the theoretical computation times of the forecasting methods and that different average computation times can be compared to evaluate the forecasting methods.

4.2. Parameter comparison

Fig. 4 shows the average forecast accuracy of the locally constant (LC) and the locally linear (LL) $k$nn models for different parameter configurations in terms of the Mean-RO-sMAPE$_{T,T+6}^{1,12}$. The most significant result is that the LL models with high regularization, i.e. low parameter values of the regularization parameter in case of PCR and PLSR or high values of the regularization parameter in case of RR and LASSO, achieve a higher mean forecast accuracy than LL models with low regularization. Moreover, the two LC models achieve a comparable performance to the LL models with high regularization. This is intuitive since a high regularization forces the model to converge to an LC model, while a low regularization forces the model to converge to an LL OLS model, which is the worst model in the evaluation. Comparing the four different regularization methods, one can see that RR performs better than LASSO and PCR performs better than PLSR on average. While PCR shows comparable results for all parameter values, PLSR is a bit more variable and the shrinkage methods depend strongly on the choice of the regularization parameter.

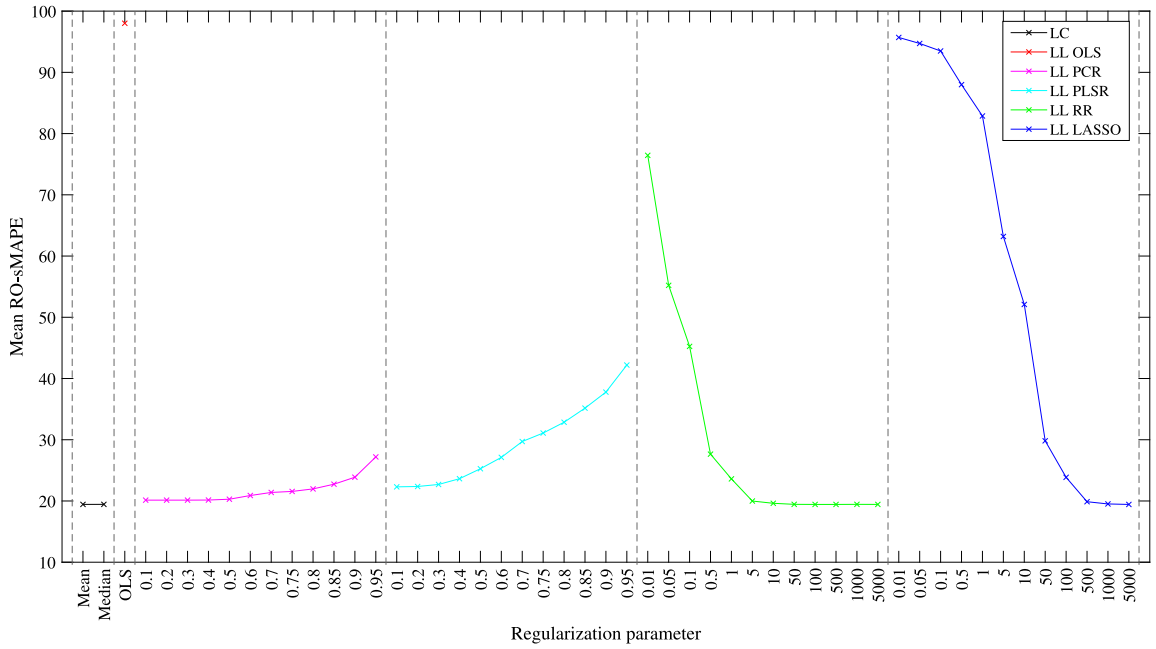


Fig. 4. Comparison of different parameter configurations of the locally constant (LC) and the locally linear (LL) k-nearest neighbor models regarding the average rolling-origin symmetric mean absolute percentage error Mean-RO-sMAPE$^{1,12}_{T,T+6}$ on the 808 considered time series of the M3 dataset.

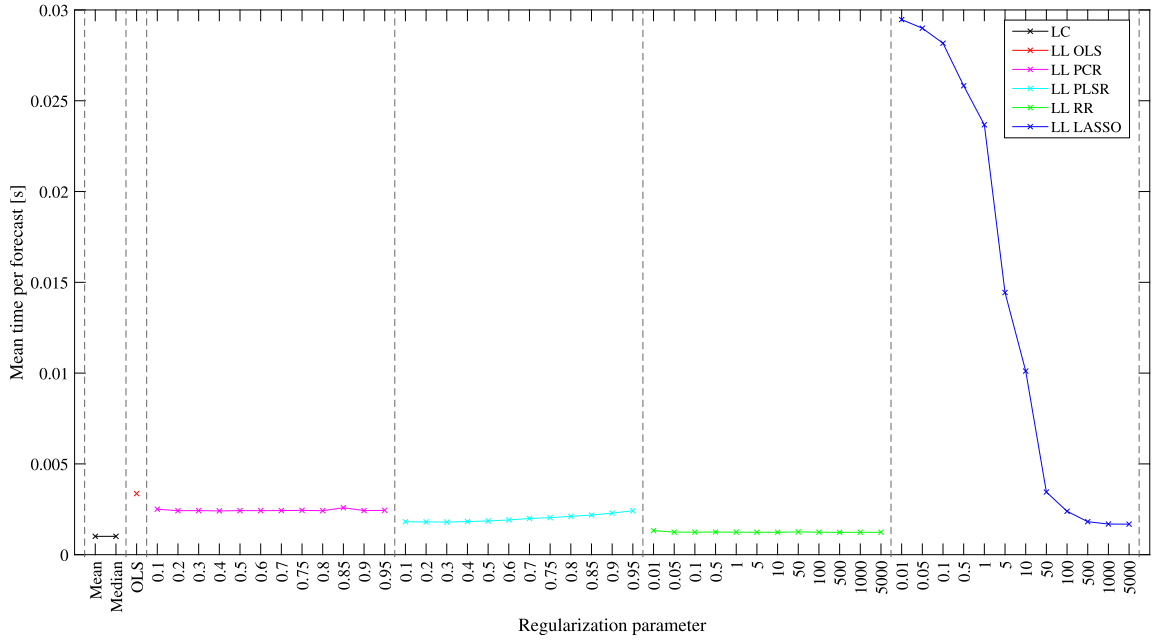


Fig. 5. Comparison of different parameter configurations of the locally constant (LC) and the locally linear (LL) k-nearest neighbor models regarding the mean time per forecast on the 808 considered time series of the M3 dataset.

Fig. 5 shows the mean times per forecast of the different parameter configurations of the LC and LL $k$nn forecasting models. It can be seen that the LC models need the shortest computation times on average. Comparing the different LL models, RR needs shorter times than PLSR, which needs shorter times than PCR. For these three regularization methods, the mean time per forecast does not vary too much with the regularization parameter. This is different for the LASSO method, which requires comparable times to the other methods for high regularization parameters and a strongly increasing mean time per forecast for decaying regularization. OLS takes long computation times but shorter than half of the LASSO models.

Table 2 merges the results of both performance criteria into one table. It is sorted by the ranks of the different models regarding mean forecast accuracy for all time series. Moreover, the table includes the results for the set of short and the set of long time series as well as the mean computation times for all time series. It can be seen that the two LC models as well as the LL RR and the LL LASSO models with high regularization achieve the lowest mean forecast errors. While the mean errors for short time series are higher than for long time series, similar models are in the range of the top-performing models. However, the ranking is slightly different. The most obvious difference can be seen in the performance of the LC Median model, which achieves the lowest mean errors for short time series but only the ninth-lowest errors

Table 2
Mean errors and mean times per forecast of the different local k-nearest neighbor forecasting models for different parameter configurations for different time series lengths.

Mean-RO-sMAPE						Model							
all		short		long		LC	LL					Mean time per forecast	
Rank	Value	Rank	Value	Rank	Value		OLS	PCR	PLSR	RR	LASSO	Value [s]	Rank
1	19.44	4	26.19	2	15.89	–	–	–	–	–	5000	0.0017	15
2	19.44	6	26.19	3	15.89	–	–	–	–	500	–	0.0012	3
3	19.44	7	26.21	1	15.89	–	–	–	–	100	–	0.0012	8
4	19.44	3	26.18	5	15.91	–	–	–	–	5000	–	0.0012	4
5	19.44	4	26.19	4	15.90	Mean	–	–	–	–	–	0.0010	1
6	19.45	1	25.99	9	16.02	Median	–	–	–	–	–	0.0010	2
7	19.45	8	26.21	6	15.91	–	–	–	–	1000	–	0.0012	6
8	19.45	2	26.17	7	15.93	–	–	–	–	50	–	0.0013	13
9	19.52	9	26.28	8	15.97	–	–	–	–	–	1000	0.0017	16
10	19.62	10	26.37	10	16.09	–	–	–	–	10	–	0.0012	5
11	19.88	16	26.96	11	16.17	–	–	–	–	–	500	0.0018	20
12	19.99	14	26.85	12	16.40	–	–	–	–	5	–	0.0012	7
13	20.14	13	26.84	13	16.62	–	–	0.3	–	–	–	0.0024	36
14	20.14	11	26.84	15	16.62	–	–	0.1	–	–	–	0.0025	41
14	20.14	11	26.84	15	16.62	–	–	0.2	–	–	–	0.0024	33
16	20.15	15	26.89	14	16.62	–	–	0.4	–	–	–	0.0024	30
17	20.29	17	27.07	17	16.73	–	–	0.5	–	–	–	0.0024	35
18	20.89	18	27.65	18	17.35	–	–	0.6	–	–	–	0.0024	34
19	21.39	19	28.10	19	17.88	–	–	0.7	–	–	–	0.0024	38
20	21.56	20	28.34	20	18.01	–	–	0.75	–	–	–	0.0024	40
21	21.97	21	28.84	21	18.37	–	–	0.8	–	–	–	0.0024	32
22	22.31	22	29.62	22	18.47	–	–	–	0.1	–	–	0.0018	19
23	22.37	23	29.70	23	18.52	–	–	–	0.2	–	–	0.0018	18
24	22.70	25	30.01	25	18.86	–	–	–	0.3	–	–	0.0018	17
25	22.74	24	29.78	26	19.05	–	–	0.85	–	–	–	0.0026	42
26	23.61	26	31.01	28	19.73	–	–	–	–	1	–	0.0012	9
27	23.64	27	31.14	27	19.70	–	–	–	0.4	–	–	0.0018	21
28	23.86	30	33.41	24	18.86	–	–	–	–	–	100	0.0024	29
29	23.88	28	31.28	29	20.00	–	–	0.9	–	–	–	0.0024	37
30	25.27	29	32.83	30	21.30	–	–	–	0.5	–	–	0.0019	22
31	27.12	32	35.35	31	22.81	–	–	–	0.6	–	–	0.0019	23
32	27.19	31	34.99	32	23.10	–	–	0.95	–	–	–	0.0024	39
33	27.65	33	35.79	34	23.38	–	–	–	–	0.5	–	0.0012	12
34	29.71	34	38.96	35	24.86	–	–	–	0.7	–	–	0.0020	24
35	29.82	37	42.54	33	23.15	–	–	–	–	–	50	0.0034	44
36	31.10	35	40.54	36	26.14	–	–	–	0.75	–	–	0.0020	25
37	32.86	36	42.52	37	27.79	–	–	–	0.8	–	–	0.0021	26
38	35.16	38	44.78	38	30.12	–	–	–	0.85	–	–	0.0022	27
39	37.78	39	47.85	39	32.50	–	–	–	0.9	–	–	0.0023	28
40	42.20	40	52.83	40	36.62	–	–	–	0.95	–	–	0.0024	31
41	45.25	41	57.17	41	38.99	–	–	–	–	0.1	–	0.0012	10
42	52.10	43	72.01	42	41.65	–	–	–	–	–	10	0.0101	45
43	55.22	42	68.52	43	48.24	–	–	–	–	0.05	–	0.0012	11
44	63.20	44	84.23	44	52.17	–	–	–	–	–	5	0.0144	46
45	76.44	45	90.51	45	69.07	–	–	–	–	0.01	–	0.0013	14
46	82.88	46	100.00	46	73.90	–	–	–	–	–	1	0.0237	47
47	87.99	47	102.60	47	80.32	–	–	–	–	–	0.5	0.0258	48
48	93.49	48	105.75	48	87.06	–	–	–	–	–	0.1	0.0282	49
49	94.72	49	106.25	49	88.67	–	–	–	–	–	0.05	0.0290	50
50	95.70	50	106.88	50	89.83	–	–	–	–	–	0.01	0.0295	51
51	98.02	51	108.10	51	92.73	–	OLS	–	–	–	–	0.0034	43

for long time series. In contrast, the LC Mean seems more robust and achieves the fourth rank for both sets. The shortest mean times per forecast are needed by the two LC models. Since the LL RR models obtain slightly lower mean forecast errors and shorter mean times per forecast than the LL LASSO models, the LC models as well as the LL RR models are chosen for the further experiments of this paper.

4.3. Model selection strategies

This subsection compares different model selection strategies for the local $k$nn forecasting models. Comparing the models of the sets LC and LL RR, Fig. 6 shows the number of time series for which a specific model achieves the highest forecast accuracy regarding RO-sMAPE$^{1,12}_{T,T+6}$. It can be seen that the LC Median obtains the lowest forecast error in approximately one fourth of all time series. However, the comparisons of the last subsection have shown that this model performs worse than some other models for long time series. In addition to the LC Median, the LC Mean as well as the LL RR models with the regularization parameters 5, 10, 50 and 100 all obtain the lowest forecast errors for more than 50 of the 808 time series. Therefore, these parameter configurations are considered as promising model candidates for model selection strategies.

Table 3 shows the mean errors and mean computation times of different individual and aggregate model selection strategies. Aggregate selection refers to selecting the same model for all time series, while an individual selection strategy performs a specific selection of a model for each time series separately. As aggregate selection methods, the six models with more than 50 wins in Fig. 6 are chosen. The individual selection strategies of this paper calculate the RO-sMAPE$^{1,12}_{t,t+6}$ of different model candidates on the validation set and select the model candidate with the lowest error. For this purpose, different sets of model candidates are compared. In addition to the six different parameter sets in Table 1, the reduced set RR$_r$ comprising the four

Table 3
Mean errors and mean times of the different individual and aggregate selection strategies for the local k-nearest neighbor forecasting models.

Mean-RO-sMAPE								Mean time per			
all		short		long				optimization		forecast	
Rank	Value	Rank	Value	Rank	Value	Set of model candidates	Number of model candidates	Value [s]	Rank	Value [s]	Rank
1	19.21	1	25.75	2	15.77	LC	2	0.20	7	0.0010	2
2	19.23	2	25.87	1	15.74	LC +RR_r	6	0.66	10	0.0011	4
3	19.44	7	26.21	3	15.89	RR100	1	0.01	3	0.0012	10
4	19.44	6	26.19	4	15.90	LC Mean	1	0.01	6	0.0010	1
5	19.45	3	25.99	6	16.02	LC Median	1	0.01	5	0.0010	3
6	19.45	5	26.17	5	15.93	RR50	1	0.01	1	0.0013	11
7	19.59	9	26.37	7	16.04	RR_r	4	0.47	9	0.0012	9
8	19.62	8	26.37	8	16.09	RR10	1	0.01	2	0.0012	6
9	19.70	4	26.12	9	16.33	LC+RR	14	1.60	12	0.0011	5
10	19.99	11	26.85	10	16.40	RR5	1	0.01	4	0.0012	7
11	20.18	10	26.75	11	16.74	RR	12	1.41	11	0.0012	8
12	20.31	12	26.92	12	16.85	LC+LL	51	21.25	16	0.0016	12
13	20.39	13	27.07	13	16.89	LASSO	12	14.57	15	0.0019	14
14	20.88	14	27.45	14	17.43	PCR	12	2.61	14	0.0024	15
15	23.88	15	31.53	15	19.87	PLSR	12	2.17	13	0.0018	13
16	98.02	16	108.10	16	92.73	OLS	1	0.30	8	0.0034	16

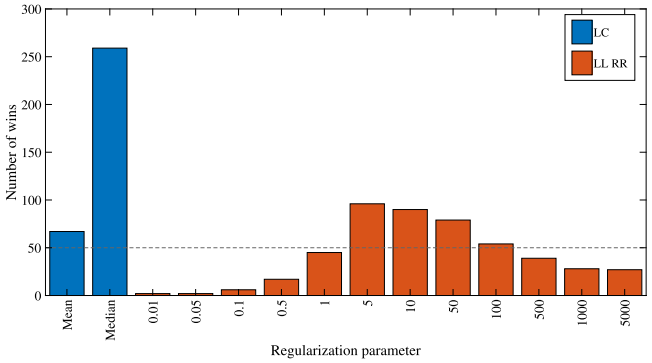


Fig. 6. Number of time series for which a specific model in a comparison of locally constant (LC) and locally linear ridge regression (LL RR) models achieves the highest forecast accuracy regarding RO-sMAPE$^{1,12}_{T,T+6}$.

parameter configurations of LL RR with more than 50 wins in Fig. 6 as well as the set LC+RR_r comprising the two LC models and the four models in RR_r are compared in Table 3.

The table shows that a selection based on the reduced set RR_r achieves higher forecast accuracy with shorter computation times than using the complete set RR. The three different individual selection strategies with the lowest mean forecast errors are LC, LC+RR_r and RR_r. These selection strategies are compared in the empirical experiments in the next sections. Since the LC Mean achieves high forecast accuracy and the shortest overall computation times, it is also compared in the empirical experiments.

5. Empirical study 1: Forecast errors and computation times

This section evaluates the performance of the four model selection strategies for local $k$nn forecasting models determined in the last section. After an experimental description, the section comprises evaluations regarding forecast errors as well as times per forecast and per optimization.

5.1. Experiment description

5.1.1. Forecasting methods

After determining LC, LC+RR_r, RR_r and LC Mean as valuable sets of model candidates for model selection strategies of the local $k$nn forecasting models in the last section, this section compares their performance against common benchmarks. For this purpose, an autoregressive integrated moving average (ARIMA) model, two different neural network forecasting methods as well as a non-seasonal and a seasonal random walk are applied. Most of the benchmark forecasts are computed using functions of the forecast package in R (Hyndman et al., 2018). The ARIMA forecasts are computed by the `auto.arima` function (Hyndman and Khandakar, 2008). The non-seasonal random walk (RW) forecasts are computed by the `naive` function and the seasonal random walk (sRW) forecasts by the `snaive` function. The first neural network approach (NNETAR) applies the `nnetar` function to build an autoregressive neural network with a decay parameter of 0.2. The second neural network approach (NNFOR) applies the `mlp` function of the NNFOR package in R (Kourentzes, 2017) to build a multilayer perceptron neural network. Both neural network approaches build ensembles of 20 networks and compute the forecasts as the mean values of the forecasts of the 20 networks. It has to be noted that all forecast values of this paper are rounded to positive integer values since the industrial time series only contain real integer quantities.

5.1.2. Dataset and performance criteria

Just as in Section 4, the 808 monthly time series with the categories industry and micro of the M3 dataset are used. The same divisions into training and test sets are applied. For more information about the dataset as well as rolling-origin forecast evaluation, see Section 4.1. In addition to the evaluation regarding the Mean-RO-sMAPE for $h_{\min} = 1$, $h_{\max} = H = 12$, $o_{\min} = T$ and $o_{\max} = T + 6$, which has been applied as the only forecast error measure in the last section, this section considers several other experiment configurations regarding different forecast origins and horizons to achieve a comprehensive evaluation of forecast accuracy. Furthermore, all experiments are conducted for the subsets of 278 short and 530 long time series as well as for the complete set of 808 time series. Moreover, this section also applies statistical tests to compare the forecast performance of the different methods. In order to compare the performances of more than two methods over many time series, multiple comparison tests have to be applied. Since the assumptions of the repeated-measures analysis of variance, such as the samples being drawn from normal distributions, are most probably violated, non-parametric tests have to be conducted. For this purpose, a Friedman test is applied to determine whether the performances of the methods differ significantly (Friedman, 1937, 1940). In case of a rejection of the null hypothesis of all forecasting methods achieving forecast accuracies with no significant distances, the Nemenyi post-hoc test is applied to find the significant differences in performance (Nemenyi, 1962; Demšar, 2006). Friedman and Nemenyi tests are non-parametric tests that determine a performance ranking of the applied forecasting methods in terms of the forecast error for each

time series. They use the mean ranks averaged over all time series as performance criterion. The performances of two or more methods are assumed to be significantly different if the difference of their mean ranks is higher than a critical distance determined by the Nemenyi test. The results of the Nemenyi tests of this paper as well as the corresponding plots have been computed in Matlab (Kourentzes, 2012).

In addition to forecast accuracy, this section again compares the forecasting methods regarding computation times per optimization and per forecast. Moreover, rankings are built for each time series and Friedman and Nemenyi tests are conducted to find significant differences.

5.2. Evaluation regarding forecast errors

Table 4 shows the Mean-RO-sMAPEs of the local $k$nn models as well as the benchmark methods for different origins, horizons and time series lengths. In addition, the results of accuracy evaluations regarding mean absolute scaled errors (MASE) can be found in Table 7 in the Appendix, showing similar results. For all experiments, it holds $o_{\min} = T$.

In case of short time series, the local $k$nn models achieve the lowest mean errors for all experiments. In most cases, the individual selection method LC, which chooses between the LC Mean and the LC Median, achieves the best results. The other three local $k$nn approaches obtain similar but slightly higher errors on average. In most cases, LC+RR$_r$ performs second-best, LC Mean performs third-best and RR$_r$ performs fourth-best. However, for long forecast horizons, the LL models seem to perform better than the LC models. ARIMA, RW and sRW obtain mean errors that are comparable to each other for most of the experiments. NNETAR does not seem to be able to find suitable parameters for short time series and therefore, this method has higher errors than these five methods for the most experiments, however still lower errors than NNFOR.

For long time series and short to medium-length horizons up to $h_{\max} = 12$ as well as for the fixed origin evaluation with $h_{\min} = 1$ and $h_{\max} = 18$, ARIMA provides the forecasts with the lowest Mean-RO-sMAPEs. For medium-length to long time series, sRW achieves the lowest mean errors. RW is the worst method, except for NNFOR in most cases. This shows that for long time series, the local $k$nn models as well as ARIMA and NNETAR seem to find enough structures in the training data to provide better forecasts than the simple benchmark method RW. Moreover, the good performance of sRW in contrast to RW indicates that many of the long time series have a seasonal behavior. The local $k$nn models obtain results that are mostly comparable to NNETAR and sRW but worse than ARIMA.

Regarding the complete set of 278 short and 530 long monthly industrial time series that are used in this paper, the local $k$nn models and ARIMA achieve the lowest errors for short to medium-length horizons. With growing horizons, the errors of ARIMA increase while the errors of the local $k$nn models and sRW are quite stable. Hence, the ranking changes. All in all, ARIMA provides forecasts with slightly lower Mean-RO-sMAPEs than the local $k$nn models and clearly lower than sRW for short horizons, while the local $k$nn models and sRW obtain lower errors for medium-length to long horizons. It has to be noted that the average fixed-origin sMAPE with $h_{\min} = 1$ and $h_{\max} = 18$ (Mean-RO-sMAPE$^{1,18}_{T,T}$) was the main accuracy criterion of the original M3-Competition. Considering the 808 monthly industrial time series and the nine forecasting methods used in this paper, the local $k$nn approaches achieve the lowest mean errors regarding this accuracy measure.

Table 5 shows the mean ranks of the forecasting methods obtained after performing Friedman and Nemenyi tests with significance levels of 0.05. In this context, the accuracy of methods is significantly different if the difference of the mean ranks is higher than the critical distance (CD) determined by the Nemenyi test. For each row, all methods with values shown in bold are the methods that are not significantly worse than the best method for this experiment. Figs. 7 and 8 show visualizations of the Nemenyi test results. Methods whose accuracies are not significantly different are connected by vertical lines. In general, the results of the Nemenyi tests are similar to the results regarding the Mean-RO-sMAPEs. However, there are some differences between these evaluations since a Nemenyi test only considers the ranking of different forecasting methods per time series but not the effect size of the errors and also not the amount of the difference between the errors that the methods achieve per time series. This means that two methods only have the same rank for a specific time series if they have exactly the same error. Considering short, long as well as all time series, again, NNFOR provides the worst results in all evaluations.

For short time series, the local $k$nn approaches achieve the lowest mean ranks and hence the best results, which is in accordance with the evaluation regarding the Mean-RO-sMAPEs. For short horizons, they obtain significantly better ranks than the other methods. For medium-length to long horizons, they still rank among the best methods but not necessarily significantly better. ARIMA is often among the best methods. NNETAR, RW and sRW obtain similar results to each other.

In case of long time series, ARIMA achieves the lowest mean ranks for all horizons and is significantly better than all other methods in most cases. The local $k$nn models provide statistically equivalent results to NNETAR and sRW in general, while RW has the worst ranks except for NNFOR in most of the cases and equivalent ranks to the other methods in some cases.

Considering the complete set of 808 short and long time series, ARIMA obtains the lowest mean ranks for most of the rolling-origin evaluations. In many cases, it is significantly better than all other methods. The local $k$nn models are generally among the second-best methods and they achieve comparable mean ranks to NNETAR and sRW.

Comparing the results regarding the Mean-RO-sMAPEs and the mean ranks in the Nemenyi tests, ARIMA provides better results considering the mean ranks than considering the mean errors. This shows that ARIMA achieves the lowest errors for most time series and most experiments across different horizons. However, for instance, the lower Mean-RO-sMAPEs of the local $k$nn models for the fixed-origin evaluation with $h_{\min} = 1$ and $h_{\max} = 18$ in contrast to the lower mean rank of ARIMA for this experiment indicates that the local $k$nn models achieve slightly higher errors than ARIMA for many cases but considerably lower errors for some cases. Thus, it can occur that one method obtains a lower mean error while another method achieves a lower mean rank.

5.3. Evaluation regarding computation times

Table 6 shows the mean times per optimization and per forecast of the different forecasting models on the 808 considered time series. A conduction of 5 replications of all experiments resulted in average standard deviations of 0.02s for the times per optimization and 0.0001s for the times per forecast. Therefore, the values were rounded to two and four decimal places respectively. Regarding times per parameter optimization, RW and sRW are the most comfortable methods since they do not require any parameter optimization time. Considering the other methods, the LC Mean needs the shortest times per optimization. The needed optimization times of NNETAR are approximately 12 times as long as the times of the LC Mean. The ARIMA method as well as the NNFOR method even need 297 and 446 times as long as the LC Mean. The three individual local $k$nn model selection strategies need higher times per optimization than NNETAR but significantly shorter times than ARIMA and NNFOR.

Considering the mean times per forecast, Table 6 shows similar results. RW requires the shortest time per forecast, followed closely by sRW. The local $k$nn approaches need the third- to sixth-shortest time, approximately 3 to 4 times as long as RW. ARIMA, NNETAR and NNFOR require longer times per forecast, approximately 13, 41 and 107 times as long as RW.

Table 4
Mean rolling-origin sMAPEs of the different local *k*-nearest neighbor forecasting models as well as benchmark methods for different origins, horizons and time series lengths (best methods are shown in bold)

Experiment configuration				Mean-RO-sMAPE								
o_{max}	h_{min}	h_{max}	Set	LC	RR_r	LC+RR_r	LC Mean	ARIMA	NNFOR	NNETAR	RW	sRW
T+17	1	1	short	**24.57***	25.11	24.81	24.66	28.15	44.44	31.53	28.27	28.78
T+17	1	1	long	13.60	13.73	13.62	13.68	**12.06***	18.71	14.67	14.10	16.01
T+17	1	1	all	**17.38***	17.64	17.47	17.46	17.60	27.56	20.47	18.97	20.40
T+15	1	3	short	**24.73***	25.21	24.95	24.87	27.90	48.48	30.94	28.12	28.70
T+15	1	3	long	14.59	14.81	14.67	14.70	**12.97***	22.78	15.56	16.09	15.93
T+15	1	3	all	**18.08***	18.39	18.21	18.20	18.11	31.62	20.85	20.23	20.33
T+12	4	6	short	**25.80***	26.26	25.86	26.13	28.25	58.14	31.69	28.37	28.69
T+12	4	6	long	15.80	16.20	15.87	16.03	**14.56***	30.75	16.86	20.08	15.95
T+12	4	6	all	**19.24***	19.66	19.31	19.51	19.27	40.18	21.96	22.93	20.33
T+9	7	9	short	**26.70***	27.47	26.78	27.36	28.99	64.36	32.39	29.13	28.91
T+9	7	9	long	16.47	16.77	16.43	16.63	**15.70***	33.28	17.36	20.67	15.82
T+9	7	9	all	**19.99***	20.45	19.99	20.32	20.28	43.97	22.53	23.58	20.32
T+6	10	12	short	**28.21***	28.91	28.21	29.06	30.06	69.20	33.25	29.83	29.45
T+6	10	12	long	17.02	17.53	16.94	17.35	16.65	32.45	18.25	17.55	**15.96***
T+6	10	12	all	20.87	21.45	20.82	21.38	21.26	45.10	23.41	21.77	**20.60***
T+3	13	15	short	**29.84***	30.32	29.90	30.41	33.55	75.17	34.69	31.06	33.25
T+3	13	15	long	19.06	19.67	19.12	19.76	19.52	38.30	20.68	19.24	**18.69***
T+3	13	15	all	**22.77***	23.33	22.83	23.42	24.34	50.99	25.50	23.30	23.70
T	16	18	short	30.47	30.59	**30.47***	30.99	35.20	76.97	34.85	33.54	33.44
T	16	18	long	20.01	20.52	19.98	20.60	20.91	38.90	21.15	23.10	**18.88***
T	16	18	all	23.61	23.99	**23.59***	24.17	25.83	52.00	25.86	26.69	23.89
T+6	1	12	short	**25.75***	26.37	25.87	26.19	28.30	58.92	30.88	28.67	28.50
T+6	1	12	long	15.77	16.04	15.74	15.90	**14.89***	29.83	16.69	18.79	15.96
T+6	1	12	all	**19.21***	19.59	19.23	19.44	19.51	39.84	21.57	22.19	20.28
T	1	18	short	26.75	27.10	**26.70***	27.32	30.69	62.31	31.34	31.70	30.05
T	1	18	long	17.25	17.55	17.16	17.56	**16.72***	29.44	17.89	19.09	16.88
T	1	18	all	20.52	20.84	**20.44***	20.92	21.53	40.75	22.51	23.43	21.41

Table 5
Mean ranks of the different local *k*-nearest neighbor forecasting models as well as benchmark methods for different origins, horizons and time series lengths (CD denotes the critical distance for significance in a Nemenyi test with significance level 0.05 and the methods that are not significantly worse than the best method are shown in bold)

Experiment configuration				CD	Mean rank regarding RO-sMAPE								
o_{max}	h_{min}	h_{max}	Set		LC	RR_r	LC+RR_r	LC Mean	ARIMA	NNFOR	NNETAR	RW	sRW
T+17	1	1	short	0.7205	**3.7662**	**4.0863**	**3.9478**	**3.7320***	4.5558	7.9137	5.6619	5.6169	5.7194
T+17	1	1	long	0.5218	5.0453	5.0519	5.1396	5.1047	**2.6566***	5.8547	4.8509	5.3472	5.9491
T+17	1	1	all	0.4226	4.6052	4.7197	4.7296	4.6324	**3.3100***	6.5631	5.1300	5.4400	5.8700
T+15	1	3	short	0.7205	**3.7338**	**4.1978**	**4.0378**	**3.6637***	4.5414	7.9388	5.4245	5.8004	5.6619
T+15	1	3	long	0.5218	4.9000	4.9236	4.9906	4.9745	**2.6623***	7.0340	4.6811	5.5811	5.2528
T+15	1	3	all	0.4226	4.4988	4.6739	4.6627	4.5235	**3.3088***	7.3453	4.9369	5.6566	5.3936
T+12	4	6	short	0.7205	**4.0324**	**4.2698**	**4.1924**	**4.0198***	**4.4838**	7.7302	5.2014	5.7932	5.2770
T+12	4	6	long	0.5218	4.6679	4.9726	4.7774	4.7123	**3.1509***	7.4170	4.5698	6.2528	4.4792
T+12	4	6	all	0.4226	4.4493	4.7308	4.5761	4.4740	**3.6095***	7.5248	4.7871	6.0947	4.7537
T+9	7	9	short	0.7205	**4.0755***	**4.5252**	**4.1565**	**4.3183**	**4.5126**	7.5935	5.2266	5.4874	5.1043
T+9	7	9	long	0.5218	4.7302	5.0462	4.7830	4.8708	**3.4453***	7.1642	4.4811	6.2943	4.1849
T+9	7	9	all	0.4226	4.5050	4.8670	4.5675	4.6807	**3.8125***	7.3119	4.7376	6.0167	4.5012
T+6	10	12	short	0.7205	**4.3561**	**4.4460**	**4.3004***	**4.7428**	**4.3471**	7.4712	5.1978	5.2860	**4.8525**
T+6	10	12	long	0.5218	4.9642	5.2311	4.9547	5.1745	**3.6736***	6.8981	4.7547	5.2679	**4.0811**
T+6	10	12	all	0.4226	4.7550	4.9610	4.7296	5.0260	**3.9053***	7.0953	4.9072	5.2741	4.3465
T+3	13	15	short	0.7205	**4.3165**	**4.4640**	**4.3040***	**4.4155**	**4.6781**	7.3741	5.3022	**4.7500**	5.3957
T+3	13	15	long	0.5218	**4.6321**	5.0896	4.7491	5.0670	**4.2000***	6.8075	4.8302	4.9151	**4.7094**
T+3	13	15	all	0.4226	**4.5235**	4.8744	**4.5959**	4.8428	**4.3645***	7.0025	4.9926	4.8583	4.9455
T	16	18	short	0.7205	**4.3669**	**4.4209**	**4.2986***	**4.5486**	**4.9694**	7.0791	5.2518	**4.9209**	5.1439
T	16	18	long	0.5218	**4.7774**	**4.8689**	**4.8009**	**4.9066**	**4.5151***	6.4340	**4.7585**	5.2142	**4.7245**
T	16	18	all	0.4226	**4.6361**	**4.7147**	**4.6281***	**4.7834**	**4.6714**	6.6559	**4.9282**	5.1132	**4.8688**
T+6	1	12	short	0.7205	**3.9748***	**4.3381**	**4.1133**	**4.0558**	**4.3903**	7.8633	5.1978	5.7608	5.3058
T+6	1	12	long	0.5218	4.7038	4.9387	4.7585	4.7519	**3.1755***	7.3245	4.5019	6.2340	4.6113
T+6	1	12	all	0.4226	4.4530	4.7321	4.5365	4.5124	**3.5934***	7.5099	4.7413	6.0712	4.8502
T	1	18	short	0.7205	**4.0845***	**4.2662**	**4.1331**	**4.2752**	5.0018	7.4748	5.1187	5.1709	5.4748
T	1	18	long	0.5218	4.8811	5.0972	4.8415	5.0991	**3.7434***	6.7660	4.7094	5.1906	4.6717
T	1	18	all	0.4226	4.6071	4.8113	**4.5978**	4.8156	**4.1764***	7.0099	4.8502	5.1838	4.9480

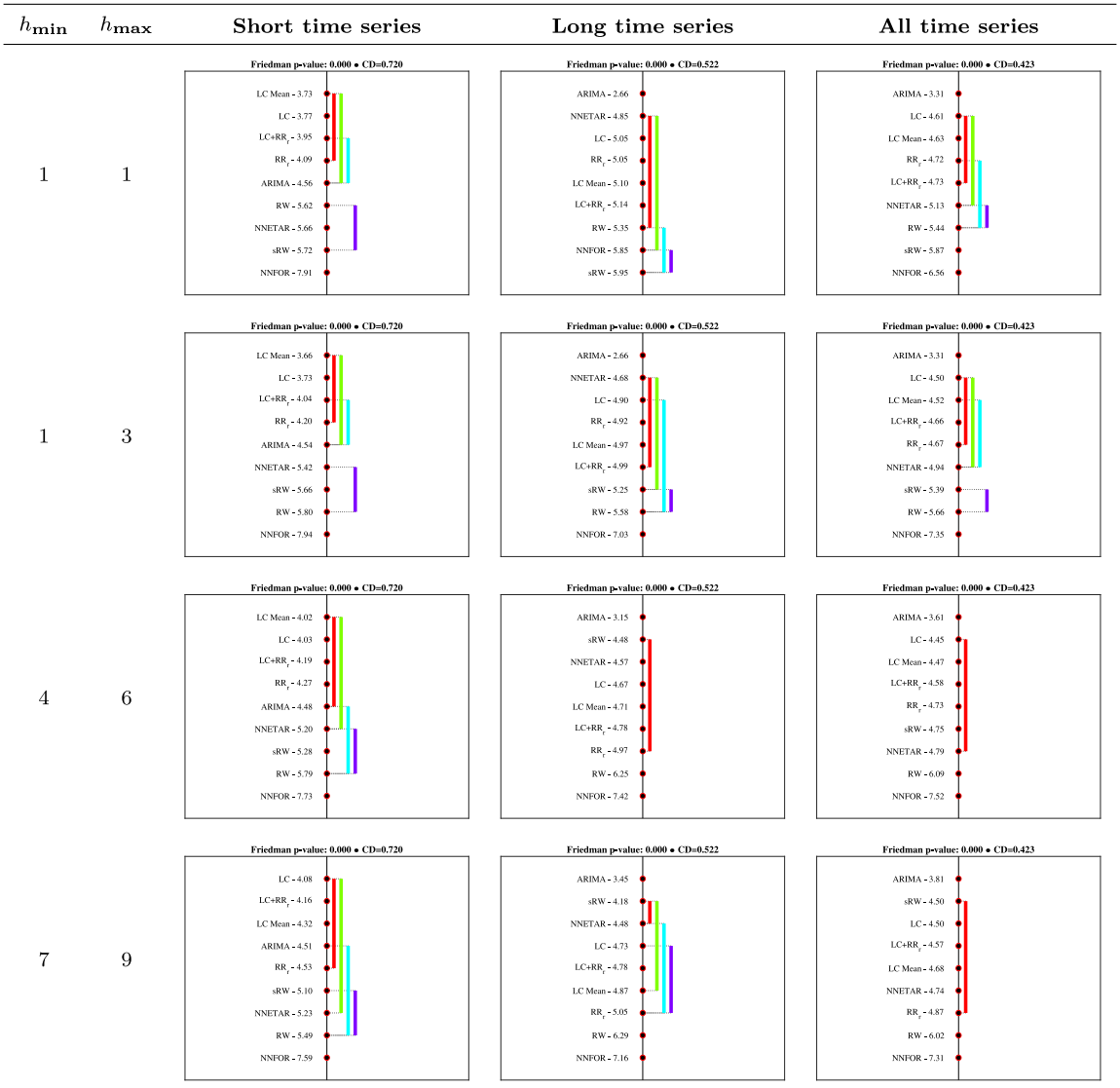


Fig. 7. Nemenyi test plots regarding the RO-sMAPEs of the different forecasting methods for short, long and all time series for short to medium-length horizons and a significance level of 0.05.

Table 6
Mean times per optimization and per forecast of the local k-nearest neighbor forecasting approaches as well as the benchmark methods (the methods with the shortest computation times are shown in bold)

Mean time per	LC	RR$_r$	LC+RR$_r$	LC Mean	ARIMA	NNFOR	NNETAR	RW	sRW
Optimization [s]	0.20	0.47	0.66	0.01	2.97	4.46	0.12	**0***	**0***
Forecast [s]	0.0010	0.0012	0.0011	0.0010	0.0038	0.0320	0.0123	**0.0003***	0.0006

Fig. 9 visualizes the results of the Friedman and Nemenyi tests regarding the times per optimization and the times per forecast for significance levels of 0.05. In both cases, the Friedman tests reject the null hypotheses of all forecasting methods needing computation times with no significant distances. Moreover, the results of the Nemenyi tests show the same rankings as Table 6. Considering the times per optimization, RW and sRW achieve the same mean rank of 1.5 indicating that they share the first rank for all time series, which makes sense since they do not require any parameter optimization. Most of the differences are statistically significant.

Regarding the times per forecast, Fig. 9 shows that RW needs shorter times than sRW. LC Mean and LC need shorter times than LL+RR$_r$, which is followed by RR$_r$, ARIMA, NNETAR and NNFOR. Almost all differences regarding times per forecast are significantly different.

6. Empirical study 2: Inventory simulation

After evaluating the performance of the forecasting methods in terms of forecast accuracy and computation times, this section evaluates their performance in inventory simulations. For this purpose, a discrete-event simulation model has been implemented. For a given time series of customer demands, each of the forecasting methods is applied to forecast the demands and a periodic order-up-to policy is used to calculate product replenishment orders. Section 6.1 describes the simulation model and the order policy. Section 6.2 defines the experiment configuration and the performance indicators. Section 6.3 shows the results of the inventory simulations.

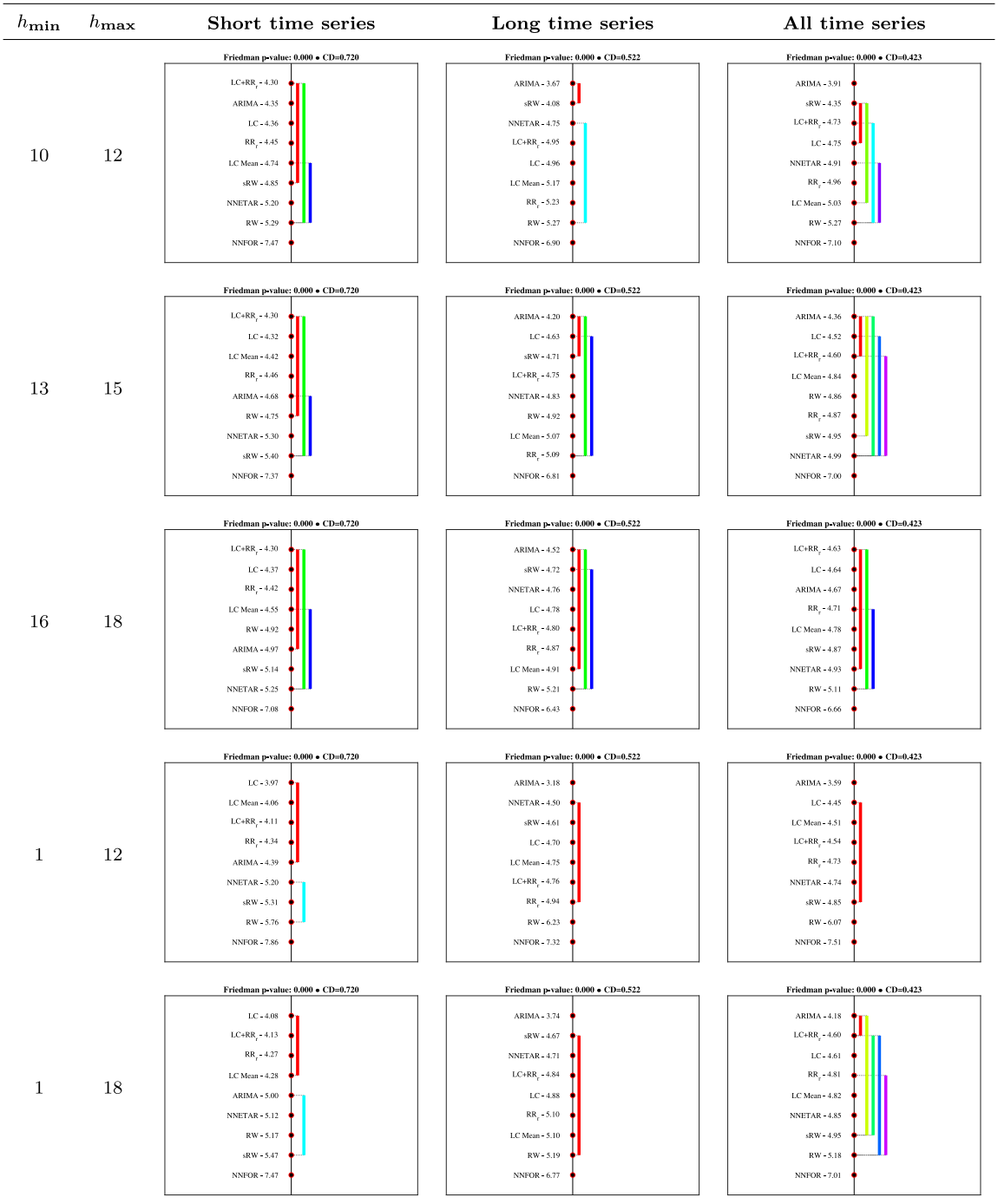


Fig. 8. Nemenyi test plots regarding the RO-sMAPEs of the different forecasting methods for short, long and all time series for medium-length to long horizons and a significance level of 0.05.

6.1. Inventory simulation model and order policy

Fig. 10 illustrates the utilized inventory model of this paper. Each of the $P = 808$ given monthly industry time series is considered as a series of monthly demands for a specific product of a company. Moreover, for each of the P products, there is one unique production line. The different production lines as well as the different demand time series are considered as mutually independent. Raw material and production capacity in terms of machines and staff are assumed to be available and are not further considered. Moreover, deliveries of finished products or raw material as well as fixed costs for setup or ordering are also neglected. This means that for each of the time series, the replenishment process of the finished goods inventory can be regarded as an independent single-stage single-item inventory control problem with the goal of achieving high service levels with low inventory levels in terms of safety stocks.

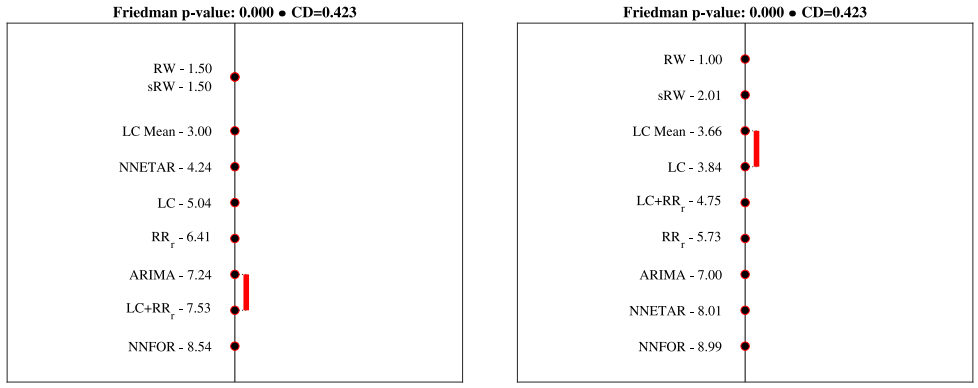
The net inventory of finished goods at the end of period t is defined as the difference of physical inventory on hand and backorders

$$i_t = i_t^+ - i_t^-, \tag{24}$$

with

$$i_t^+ = \max(i_t, 0) \tag{25}$$

$$i_t^- = \max(-i_t, 0). \tag{26}$$



(a) Time per optimization (b) Time per forecast

Fig. 9. Visualization of the Nemenyi tests for significance levels of 0.05 regarding the computation times per optimization (a) and per forecast (b) of the different forecasting methods on the 808 considered time series of the M3 dataset (CD denotes the critical distance for significance in a Nemenyi test with significance level 0.05).

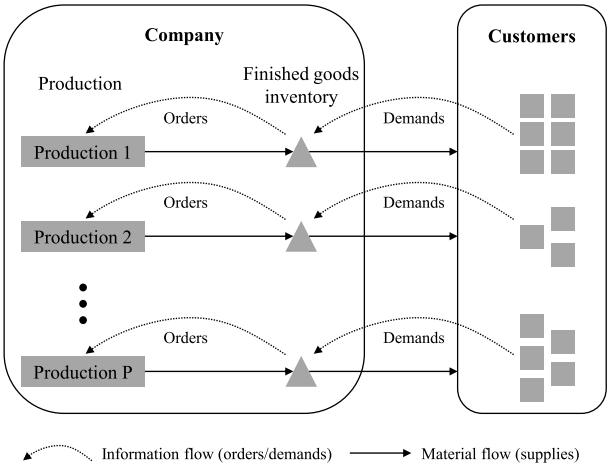


Fig. 10. Inventory model.

It is assumed that if demands cannot be satisfied because the physical inventory on hand is zero, all open demands are backlogged and can be satisfied in one of the following periods. It is furthermore assumed that the production of new products takes a lead time of L time periods, which means that the products p_t ordered at the end of period t will be available L periods later, i.e. at the beginning of period $t + L + 1$. In this context, the lead time as well as the ordered quantities can be regarded as reliable and no uncertainty is associated with these values. The ordered products that are not available yet at the end of period t are called the work-in-process

$$w_t = \begin{cases} 0, & \text{if } L = 0 \\ \sum_{j=1}^{L} p_{t-j}, & \text{else.} \end{cases} \tag{27}$$

Using these values, the inventory position at the end of period t is defined as the sum of net inventory and work-in-process

$$I_t = i_t + w_t. \tag{28}$$

For inventory replenishment, this paper considers an (r, S_t) periodic-review order-up-to policy with a review interval $r = 1$ and a dynamic order-up-to level S_t. This means that the sequence of events of the system is as follows: (i) arrival of ordered products, (ii) satisfaction of customer demands from the finished goods inventory or backlogging, (iii) update of net inventory, work-in-process and inventory position, (iv) calculation of demand forecasts for the following periods, (v) calculation of dynamic order-up-to level and (vi) ordering new products. These events are now described in detail.

At the beginning of a period t, the products p_{t-L-1} ordered L periods earlier, i.e. at the end of period $t - L - 1$, are completed and added to the finished goods inventory on hand i_t^+. After satisfying possible backorders, the customer demands y_t, which occur in the course of the period, are satisfied by taking the amount of products from the inventory of finished goods as long as $i_t^+ > 0$. If $i_t^+ = 0$, the demands that cannot be satisfied are backlogged. At the end of the period, the complete demand y_t of the tth period is known and the net inventory, the work-in-process and the inventory position are updated as

$$i_t = i_{t-1} + p_{t-L-1} - y_t \tag{29}$$

$$w_t = w_{t-1} + p_{t-1} - p_{t-L-1} \tag{30}$$

$$I_t = I_{t-1} + p_{t-1} - y_t. \tag{31}$$

Subsequently, the forecasts of the customer demands of the following periods $\hat{y}_{t,t+h}$ for $h = 1, \ldots, L$ are calculated by applying the selected forecasting method based on the time series of past customer demands. For a predefined target service level α^*, the dynamic order-up-to level S_t shall be chosen so that the probability that all customer demands within the next $L + 1$ periods will be satisfied immediately is α^*. It is calculated as

$$S_t = \sum_{h=1}^{L+1} \hat{y}_{t,t+h} + s_t, \tag{32}$$

where s_t denotes the dynamic safety stock at the end of period t. The safety stock shall account for the forecast uncertainty in terms of lead time forecast errors. Let E_L be the random variable of lead time forecast errors measured as mean squared error of the lead time forecasts. The safety stock is defined as the quantity of products that the lead time forecast error will not exceed with probability α^*, which means

$$\Pr(E_L \leq s_t) \geq \alpha^*. \tag{33}$$

Furthermore, let $F_{E_L}^{-1}(\cdot)$ be the inverse distribution function of the lead time errors, then the dynamic safety stock at period t can be estimated as the target service level quantile

$$s_t = Q_{E_L}(\alpha^*) = F_{E_L}^{-1}(\alpha^*). \tag{34}$$

In this context, a common assumption used in many theoretical and practical applications is that forecast errors are i.i.d. normal random variables. However, this assumption is often violated since forecast errors are correlated over the lead time even if demands are not (Prak et al., 2017) and their distribution may deviate from normality (Manary et al., 2009; Lee, 2014). Moreover, recent research has shown that

traditional safety stock estimation methods based on the assumption of i.i.d. normally distributed errors may lead to significantly underestimated safety stocks if the assumption is violated (Prak et al., 2017). Therefore, this paper uses the non-parametric approach of kernel density estimation (Silverman, 1986) to estimate the distribution of lead time forecast errors, which has been suggested in Manary et al. (2009) and Trapero et al. (2019). For a given sample of past lead time forecast errors $e_{L,1}, \dots, e_{L,J}$, the true probability density function $f_{E_L}(\cdot)$ is estimated as

$$\hat{f}_{E_L}(e) = \frac{1}{JB} \sum_{j=1}^{J} K\left(\frac{e - e_{L,j}}{B}\right), \tag{35}$$

where J is the sample size, B is the bandwidth and $K(\cdot)$ is the kernel smoothing function, which integrates to 1. As suggested by Silverman (1986) and Trapero et al. (2019), the Epanechnikov kernel

$$K_{\text{Epanechnikov}}(u) = \begin{cases} \frac{3}{4\sqrt{5}}\left(1 - \frac{1}{5}u^2\right), & -\sqrt{5} \le u \le \sqrt{5} \\ 0, & \text{else} \end{cases} \tag{36}$$

is used and the bandwidth is chosen as

$$B_{\text{opt}} = 0.9AN^{-\frac{1}{5}}, \tag{37}$$

where A is an adaptive value of spread defined as

$$A = \min(\text{standard deviation}, \text{interquartile range}/1.34). \tag{38}$$

Using the estimated probability density function, the safety stock is calculated by (34) and the order-up-to level by (32). Finally, a quantity p_t of new products is ordered at the end of period t so that the inventory position is raised to the order-up-to level, which means

$$p_t = \max(S_t - I_t, 0). \tag{39}$$

It is assumed that no products are ordered if the inventory position at the end of a period is higher than the dynamic order-up-to level. At the beginning of the following period, the same sequence of events is conducted again.

The inventory simulation model of this paper can be regarded as a discrete-event simulation model (Banks et al., 2013; Law, 2015). Since the review interval is assumed to be $r = 1$, the system changes its state at every end of a period. Then, the complete customer demand of this period is known and the state variables of the system are updated by executing (29)–(39). The simulation model has been implemented in Matlab (The MathWorks Inc., 2017). In order to verify the correct implementation, the values of the state variables have been calculated manually for different examples of given customer demands and forecasts. Afterwards, these values have been compared to the values of the state variables calculated by the simulation model. This process was conducted using the debugger. In addition to the formal verification that a model has been correctly implemented, it has to be validated that the inventory model is representative of real inventory systems. Since single-stage single-item inventory models with the described (r, S_t) periodic-review order-up-to policy have been applied in several theoretical and practical applications and it is agreed that they represent a common behavior of real inventory systems (Lee et al., 1997; Minner, 2003; Babai and Dallery, 2009; Wang and Petropoulos, 2016; Silver et al., 2016; Petropoulos et al., 2019), it is considered as a valid model for evaluating forecasting methods regarding inventory performance.

6.2. Experiment description and performance indicators

The inventory simulations of this paper are conducted to evaluate the local $k$nn forecasting approaches in terms of the inventory metrics they obtain, when they are used in combination with the periodic review order-up-to policy described in the last subsection. Again, the dataset of 808 monthly industrial time series of the M3-Competition described in Section 4.1.2 is used. Each of the time series is considered to describe the demand for a specific product and the demands for different products are considered to be mutually independent. This means that 808 independent inventory simulations are conducted. For each simulation, one of the nine forecasting methods of this paper has to be selected and the target service level as well as the lead time have to be defined and the customer demands per month are defined by one of the 808 time series. In this paper, a lead time of $L = 1$ is assumed, which means that two-step ahead forecasts are needed in rolling-origin simulations in order to forecast the lead time demand values. As target service levels, values of 0.9, 0.95 and 0.99 are considered. Each inventory simulation is conducted for the values of the test set $\mathbf{y}_{\text{test}} = y_{T+1}, \dots, y_{T+H}$ of the time series. In order to start a simulation, initial values for the net inventory, the work-in-process and the inventory position have to calculated. For this purpose, forecasts and errors are also computed for the validation set $\mathbf{y}_{\text{val}} = \{y_{T-17}, \dots, y_T\}$. The initial value of net inventory is set as the safety stock

$$i_T = s_T = \hat{F}^{-1}_{E_L}(\alpha^*), \tag{40}$$

where $\hat{F}^{-1}_{E_L}$ is the inverse empirical distribution function estimated by kernel density estimation using the lead time forecast errors on the validation set. In order to determine an initial value for the work-in-process, past product orders have to be known, which are initialized as the average of the customer demands of the validation set:

$$p_{T-j} = \frac{1}{T} \sum_{t=0}^{17} y_{T-t}, \text{ for all } j \in \{1, \dots, L\}. \tag{41}$$

Using these values, the work-in-process, the inventory position and the dynamic order-up-to level at the end of period T are initialized by (27), (28) and (32). The production orders p_T at the end of the validation set are generated using (39). Afterwards, the simulation is conducted by iterative executions of (29)–(39). It has to be noted that all values for forecasts, inventories, safety stocks, order-up-to levels and orders are rounded to positive integers.

To evaluate the inventory performance of a forecasting method for a time series, the achieved service level α and the needed scaled safety stock s are considered. The achieved service level is defined as the ratio of periods, in which the complete customer demand is satisfied from the inventory on hand without backorders. It is scaled by the mean customer demand for the specific product. The described inventory simulations are conducted separately for each of the $P = 808$ time series and the mean performance criteria $\bar{\alpha}$ and $\bar{s}$ are calculated.

6.3. Evaluation regarding inventory metrics

This subsection evaluates the forecasting methods in terms of their inventory performance in an inventory simulation conducted according to the descriptions in the last two subsections. Fig. 11 shows trade-off curves of scaled safety stocks versus achieved cycle service levels for short, long and the complete set of all time series for different target service levels. Since the goal is to achieve a high cycle service level with low safety stock, curves that are more left as well as above others show a better performance. It can be seen that all forecasting methods need higher safety stocks for the set of short time series than for the set of long time series. For a target service level of 0.9, all methods overperform, while they are close to the target service level for 0.95 and they slightly underperform on average for a target service level of 0.99.

The local $k$nn approaches perform very similarly on average. They achieve the best trade-offs for the three different target service levels for short time series, which is in accordance with the forecast accuracy results in the last section. For the set of long as well as the complete set of short and long time series, the models also achieve good trade-offs, but ARIMA achieves higher service levels with lower safety stocks. Since RW leads to high forecast errors with a low variance, it sets very high safety stocks but it achieves high service levels.

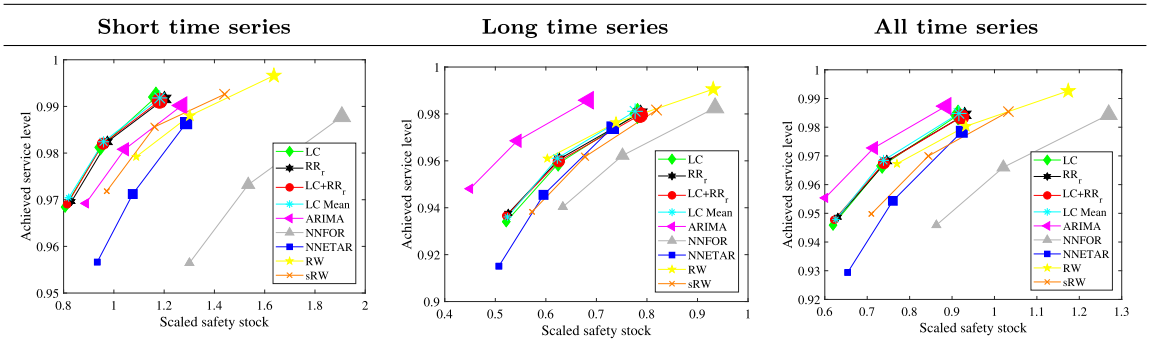


Fig. 11. Trade-off curves of achieved cycle service levels versus scaled safety stocks for the different forecasting methods for short, long and all time series and for different target service levels of 0.9, 0.95 and 0.99 (larger symbols correspond to higher target service levels).

7. Conclusion

This paper has assessed the suitability of local *k*-nearest neighbor models to forecast industrial time series data. After a thorough description of the theoretical background, an extensive parameter comparison has been conducted to find reasonable parameters and model selection strategies that lead to stable forecasting performance. In this context, the locally constant *k*nn model was tested with a local averaging according to an arithmetic mean and a median. Moreover, different regularization methods have been compared for the locally linear *k*nn models, namely ridge regression, the least absolute shrinkage and selection operator method, principal component regression as well as partial least squares regression. In the parameter comparison, the locally constant models and the locally linear *k*nn ridge regression with high regularization parameters achieved the best compromise in terms of low mean forecast errors and short computation times. Therefore, these two parameter sets of the local *k*nn forecasting models have been regarded as model candidates for individual model selection strategies. In a comparison of different sets of model candidates for model selection, the set of locally constant models, a reduced set of locally linear ridge regression models as well as a combined set of these two have achieved the most promising results and hence they have been used in the empirical studies of this paper.

In order to evaluate the forecasting performance of the local *k*nn models in an industrial context, different experiments have been conducted using the 808 monthly time series of the well-known M3-Competition with the categories industry and micro. To obtain objective measures of forecasting performance, the local *k*nn approaches have been compared to established benchmark methods, namely an ARIMA method, two automatic artificial neural network algorithms as well as a seasonal and a non-seasonal random walk. Moreover, the forecasting performance has been evaluated regarding different criteria. Firstly, the forecasting accuracy has been assessed in terms of rolling-origin symmetric mean absolute percentage errors considering horizons of different lengths and forecasts from multiple origins on the test sets of the 808 considered time series. In this context, Friedman and Nemenyi tests have been conducted in addition to calculating the mean forecast errors in order to test for statistical significance of the performance differences between the applied methods. Secondly, the computation times per parameter optimization and per forecast of the different methods have been compared, which is often neglected in other studies. Thirdly, this paper has evaluated the local *k*nn models in terms of inventory performance. For this purpose, inventory simulations have been conducted and the methods have been compared regarding the achieved service levels and the needed scaled safety stocks.

In the empirical study, the local *k*nn forecasting approaches have achieved promising results regarding the different performance criteria in multiple experiments. While the local *k*nn approaches only achieved average accuracy for long time series, they obtained the best results among all methods regarding short time series. This shows that these approaches also provide suitable forecasts in cases when the other methods would need more data to find enough structure for reasonable forecasts. For the complete set of 278 short and 530 long monthly industrial time series, the local *k*nn approaches provided a good overall performance since they ranked among the best methods for all experiments. While locally constant models achieved the best results in most cases, locally linear models offered potential to forecast at long forecast horizons. In particular, it has to be noted that the local *k*nn approaches achieved the lowest mean errors among all applied methods in this paper for the original experiment configuration of the M3-Competition, a fixed-origin evaluation of horizons 1 to 18.

After the evaluation regarding forecast accuracy, an investigation of computation times has shown that the local *k*nn approaches work efficiently. Both, mean times per parameter optimization and per forecast have been significantly shorter than for the ARIMA method. Hence, the local *k*nn models are particularly appropriate in big data contexts, where many forecasts for many time series have to be conducted at regular intervals.

In addition to forecast accuracy and computational efficiency, the inventory performance of the local *k*nn forecasting approaches has been assessed in inventory simulations. Here, the approaches again achieved very good results for short time series and average results for long time series. For the complete set of all time series, they performed slightly worse than ARIMA but better than all other applied methods.

All in all, the proposed local *k*nn forecasting approaches have achieved a high mean forecast accuracy, short computation times as well as a good inventory performance in the extensive evaluations of this paper. In addition, it has to be noted that standard approaches for the estimation of embedding parameters have been used in all evaluations of this paper. In this context, a detailed investigation regarding the selection of appropriate embedding parameters could offer the potential for further accuracy improvements. While local *k*nn models have mostly been neglected in past studies on industrial datasets, they should be considered as a promising alternative to established forecasting methods in future studies as well as in industrial practice.

CRediT authorship contribution statement

Mirko Kück: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization. **Michael Freitag:** Conceptualization, Writing - review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 7
Mean rolling-origin MASEs of the different local *k*-nearest neighbor forecasting models as well as benchmark methods for different origins, horizons and time series lengths (best methods are shown in bold)

Experiment configuration				Mean-RO-MASE								
o_{max}	h_{min}	h_{max}	Set	LC	RR_r	$LC+RR_r$	LC Mean	ARIMA	NNFOR	NNETAR	RW	sRW
T+17	1	1	short	**0.8537***	0.8708	0.8633	0.8549	1.0016	1.4678	1.2116	0.9671	0.9879
T+17	1	1	long	1.1094	1.106	1.104	1.1163	**0.931***	1.3274	1.3497	1.0913	1.4963
T+17	1	1	all	1.0214	1.0251	1.0212	1.0264	**0.9553***	1.3757	1.3022	1.0486	1.3214
T+15	1	3	short	**0.8698***	0.8845	0.878	0.8705	0.99	1.6942	1.1803	0.9675	0.989
T+15	1	3	long	1.2302	1.2426	1.232	1.2418	**1.0476***	1.745	1.4915	1.2782	1.4839
T+15	1	3	all	1.1062	1.1194	1.1102	1.114	**1.0278***	1.7275	1.3844	1.1713	1.3136
T+12	4	6	short	**0.9216***	0.9338	0.923	0.9254	1.0077	2.4036	1.2186	0.9959	0.9898
T+12	4	6	long	1.4491	1.4964	1.4546	1.4746	**1.3102***	2.6062	1.7301	1.6873	1.507
T+12	4	6	all	1.2676	1.3028	1.2717	1.2856	**1.2062***	2.5365	1.5541	1.4494	1.329
T+9	7	9	short	**0.953***	0.9789	0.9567	0.9715	1.0343	2.9714	1.2373	1.0162	0.9919
T+9	7	9	long	1.5891	1.6249	1.5874	1.6024	**1.4694***	2.9362	1.8404	1.7994	1.5135
T+9	7	9	all	1.3703	1.4026	1.3704	1.3853	**1.3197***	2.9483	1.6329	1.53	1.334
T+6	10	12	short	0.9945	1.0148	**0.9924***	1.0246	1.0553	3.4654	1.2506	1.0211	0.9938
T+6	10	12	long	1.6544	1.7198	1.6533	1.6899	1.5583	2.9968	1.9171	1.6172	**1.5062***
T+6	10	12	all	1.4274	1.4772	1.4259	1.461	1.3852	3.158	1.6878	1.4121	**1.3299***
T+3	13	15	short	1.0469	1.0599	**1.0458***	1.066	1.1477	4.0453	1.3102	1.081	1.1662
T+3	13	15	long	1.8271	1.899	1.8292	1.8985	**1.7714***	3.4792	2.1013	1.7776	1.9304
T+3	13	15	all	1.5587	1.6103	1.5596	1.6121	1.5568	3.674	1.8291	**1.5379***	1.6675
T	16	18	short	1.0738	1.0709	**1.0637***	1.0933	1.201	4.7098	1.3233	1.2174	1.168
T	16	18	long	1.9724	1.9912	**1.9483***	2.0206	1.9524	3.8122	2.1707	2.0854	1.9815
T	16	18	all	1.6632	1.6746	**1.644***	1.7016	1.6939	4.1211	1.8791	1.7868	1.7016
T+6	1	12	short	**0.9293***	0.9503	0.9346	0.9402	1.0074	2.5853	1.1844	1.0097	0.988
T+6	1	12	long	1.4702	1.5056	1.4691	1.4878	**1.3406***	2.4967	1.7023	1.5995	1.4951
T+6	1	12	all	1.2841	1.3145	1.2852	1.2994	**1.2259***	2.5272	1.5241	1.3966	1.3206
T	1	18	short	0.963	0.9705	**0.9576***	0.9804	1.0849	3.264	1.1956	1.1646	1.0481
T	1	18	long	1.6222	1.6439	1.6076	1.6531	**1.484***	2.6902	1.796	1.6509	1.6381
T	1	18	all	1.3954	1.4122	1.384	1.4217	**1.3467***	2.8877	1.5894	1.4835	1.4351

Appendix

See Table 7.

References

Abarbanel, H.D.I., Brown, R., Kadtke, J.B., 1990. Prediction in chaotic nonlinear systems: Methods for time series with broadband Fourier spectra. Phys. Rev. A 41, 1782–1807.

Adeodato, P.J.L., Arnaud, A.L., Vasconcelos, G.C., Cunha, R.C., Monteiro, D.S., 2011. MLP ensembles improve long term prediction accuracy over single networks. Int. J. Forecast. 27, 661–671.

Aggarwal, S.K., Saini, L.M., Kumar, A., 2009. Electricity price forecasting in deregulated markets: A review and evaluation. Int. J. Electr. Power Energy Syst. 31, 13–22.

Aguirre, L.A., Letellier, C., 2009. Modeling nonlinear dynamics and chaos: A review. Math. Probl. Eng. 2009, 1–35.

Al-Qahtani, F.H., Crone, S.F., 2013. Multivariate k-nearest neighbour regression for time series data - A novel algorithm for forecasting UK electricity demand. In: Proceedings of the International Joint Conference on Neural Networks. IEEE, pp. 1–8.

Alvarez-Diaz, M., 2008. Exchange rates forecasting: local or global methods? Appl. Econ. 40, 1969–1984.

Babai, M.Z., Ali, M.M., Boylan, J.E., Syntetos, A.A., 2013. Forecasting and inventory performance in a two-stage supply chain with ARIMA(0,1,1) demand: Theory and empirical analysis. Int. J. Prod. Econ. 143, 463–471.

Babai, M.Z., Dallery, Y., 2009. Dynamic versus static control policies in single stage production-inventory systems. Int. J. Prod. Res. 47, 415–433.

Banks, J., Carson, J.S., Nelson, B.L., Nicol, D.M., 2013. Discrete-Event System Simulation, fifth ed. Pearson Education Limited, p. 568.

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

Bontempi, G., Ben Taieb, S., 2011. Conditionally dependent strategies for multiple-step-ahead prediction in local learning. Int. J. Forecast. 27, 689–699.

Bontempi, G., Birattari, M., Bersini, H., 1999. Local learning for iterated time series prediction. In: Machine Learning: Proceedings of the Sixteenth International Conference. ICML'99. Morgan Kaufmann Publishers, pp. 32–38.

Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2015. Time Series Analysis: Forecasting and Control. Wiley, p. 712.

Broer, H.H.W., Takens, F., 2011. Dynamical Systems and Chaos. Springer, New York.

Cao, L., Soofi, A.S., 1999. Nonlinear deterministic forecasting of daily dollar exchange rates. Int. J. Forecast. 15, 421–430.

Casdagli, M., 1989. Nonlinear prediction of chaotic time series. Physica D 35, 335–356.

Cecen, A.A., Erkal, C., 1996a. Distinguishing between stochastic and deterministic behavior in foreign exchange rate returns: Further evidence. Econom. Lett. 51, 323–329.

Cecen, A.A., Erkal, C., 1996b. Distinguishing between stochastic and deterministic behavior in high frequency foreign exchange rate returns: Can non-linear dynamics help forecasting? Int. J. Forecast. 12, 465–473.

Chankov, S., Hütt, M.T., Bendul, J., 2016. Synchronization in manufacturing systems: quantification and relation to logistics performance. Int. J. Prod. Res. 54, 6033–6051.

Cho, S., Erkoc, M., 2009. Design of predictable production scheduling model using control theoretic approach. Int. J. Prod. Res. 47, 2975–2993.

Cleveland, W.S., 1979. Robust locally weighted regression and smoothing scatterplots. J. Amer. Statist. Assoc. 74, 829–836.

Colledani, M., Tolio, T., Fischer, A., Iung, B., Lanza, G., Schmitt, R., Váncza, J., 2014. Design and management of manufacturing systems for production quality. CIRP Ann. Manuf. Technol. 63, 773–796.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273–297.

Cover, T.M., Hart, P.E., 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory 13, 21–27.

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

Crone, S.F., Kourentzes, N., 2010. Feature selection for time series prediction - A combined filter and wrapper approach for neural networks. Neurocomputing 73, 1923–1936.

Crone, S.F., Lessmann, S., Pietsch, S., 2008. Forecasting with computational intelligence - An evaluation of support vector regression and artificial neural networks for time series prediction. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings. IEEE, pp. 3159–3166.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30.

Dimri, A.P., Joshi, P., Ganju, A., 2008. Precipitation forecast over western Himalayas using k -nearest neighbour method. Int. J. Climatol. 28, 1921–1931.

Donner, R., Scholz-Reiter, B., Hinrichs, U., 2008. Nonlinear characterization of the performance of production and logistics networks. J. Manuf. Syst. 27, 84–99.

Engster, D., 2011. Local-and Cluster Weighted Modeling for Prediction and State Estimation of Nonlinear Dynamical Systems (Ph.D. thesis). Niedersächsische Staats-und Universitätsbibliothek Göttingen.

Farmer, J., Sidorowich, J., 1987. Predicting chaotic time series. Phys. Rev. Lett. 59, 845–848.
Ferbar Tratar, L., Mojškerc, B., Toman, A., 2016. Demand forecasting with four-parameter exponential smoothing. Int. J. Prod. Econ. 181, 162–173.
Fernández-Rodríguez, F., Sosvilla-Rivero, S., Andrada-Félix, J., 1999. Exchange-rate forecasts with simultaneous nearest-neighbour methods: evidence from the EMS. Int. J. Forecast. 15, 383–392.
Fix, E., Hodges, J.L., 1951. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. Technical Report. University of California, Berkeley.
Fleischmann, B., Meyr, H., Wagner, M., 2015. Advanced planning. In: Supply Chain Management and Advanced Planning (Fourth Edition): Concepts, Models, Software, and Case Studies. Springer, Berlin, Heidelberg, pp. 71–95.
Fraser, A.M., Swinney, H.L., 1986. Independent coordinates for strange attractors from mutual information. Phys. Rev. A 33, 1134–1140.
Freitag, M., Becker, T., Duffie, N.A., 2015. Dynamics of resource sharing in production networks. CIRP Ann. Manuf. Technol. 64, 435–438.
Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Amer. Statist. Assoc. 32, 675–701.
Friedman, M., 1940. A comparison of alternative tests of significance for the problem of m rankings. Ann. Math. Stat. 11, 86–92.
Fritzsche, R., Gupta, J.N.D., Lasch, R., 2014. Optimal prognostic distance to minimize total maintenance cost: The case of the airline industry. Int. J. Prod. Econ. 151, 76–88.
Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., Helu, M., 2015. Cloud-enabled prognosis for manufacturing. CIRP Ann. Manuf. Technol. 64, 749–772.
Gardner, E.S., 2006. Exponential smoothing: The state of the art-Part II. Int. J. Forecast. 22, 637–666.
Hahn, H., Meyer-Nieberg, S., Pickl, S., 2009. Electric load forecasting methods: Tools for decision making. European J. Oper. Res. 199, 902–907.
Hastie, T., Tibshirani, R., Friedman, J., 2009. the Elements of Statistical Learning, second ed. In: Springer Series in Statistics, Springer, New York, NY, USA.
Helbing, D., 2003. Modelling supply networks and business cycles as unstable transport phenomena. New J. Phys. 5, 1–28.
Hong, T., Fan, S., 2016. Probabilistic electric load forecasting: A tutorial review. Int. J. Forecast. 32, 914–938.
Hwarng, H.B., Xie, N., 2008. Understanding supply chain dynamics: A chaos perspective. European J. Oper. Res. 184, 1163–1178.
Hyndman, R.J., 2020. A brief history of forecasting competitions. Int. J. Forecast. 36, 7–14.
Hyndman, R.J., Athanasopoulos, G., 2018. Forecasting: Principles and Practice, second ed. OTexts, Melbourne, Australia, URL www.otexts.com/fpp2.
Hyndman, R.J., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., Yasmeen, F., 2018. Forecast: Forecasting functions for time series and linear models. URL http://pkg.robjhyndman.com/forecast.
Hyndman, R.J., Khandakar, Y., 2008. Automatic time series forecasting: The forecast package for R. J. Stat. Softw. 27, 1–22.
Hyndman, R.J., Koehler, A., Ord, K., Snyder, R., 2008. Forecasting with Exponential Smoothing, first ed..
Jaditz, T., Sayers, C.L., 1998. Out-of-sample forecast performance as a test for nonlinearity in time series. J. Bus. Econom. Statist. 16, 110–117.
Jayawardena, A.W., Li, W.K., Xu, P., 2002. Neighbourhood selection for local modelling and prediction of hydrological time series. J. Hydrol. 258, 40–57.
Jursa, R., Rohrig, K., 2008. Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models. Int. J. Forecast. 24, 694–709.
Kantz, H., Holstein, D., Ragwitz, M., K. Vitanov, N., 2004. Markov Chain model for turbulent wind speed data. Physica A 342, 315–321.
Kantz, H., Schreiber, T., 2004. Nonlinear Time Series Analysis, second ed. Cambridge University Press, Cambridge.
Katzorke, I., Pikovsky, A., 2006. Chaos and complexity in a simple model of production dynamics. Discrete Dyn. Nat. Soc. 5, 179–187.
Kennel, M.B., Brown, R., Abarbanel, H.D.I., 1992. Determining embedding dimension for phase-space reconstruction using a geometrical construction. Phys. Rev. A 45, 3403–3411.
Kilger, C., Wagner, M., 2015. Demand planning. In: Stadtler, H., Kilger, C. (Eds.), Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies. Springer, Berlin, pp. 125–154.
Klug, F., 2016. Analysing bullwhip and backlash effects in supply chains with phase space trajectories. Int. J. Prod. Res. 54, 3906–3926.
Kourentzes, N., 2012. Nemenyi tests. URL http://nikolaos.kourentzes.com.
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., 2017. Nnfor: Time series forecasting with neural networks. URL https://cran.r-project.org/package=nnfor.
Kourentzes, N., Barrow, D.K., Crone, S.F., 2014. Neural network ensemble operators for time series forecasting. Expert Syst. Appl. 41, 4235–4244.
Kück, M., Becker, T., Freitag, M., 2016a. Emergence of non-predictable dynamics caused by shared resources in production networks. Procedia CIRP 41, 520–525.
Kück, M., Crone, S.F., Freitag, M., 2016b. Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data. In: Proceedings of the International Joint Conference on Neural Networks. IEEE, pp. 1499–1506.
Kück, M., Scholz-Reiter, B., 2013. A genetic algorithm to optimize lazy learning parameters for the prediction of customer demands. In: Proceedings - 2013 12th International Conference on Machine Learning and Applications. ICMLA 2013. pp. 160–165.
Kück, M., Scholz-Reiter, B., Freitag, M., 2014. Robust methods for the prediction of customer demands based on nonlinear dynamical systems. Procedia CIRP 19, 93–98.
Kugiumtzis, D., Lingjaerde, O.C., Christophersen, N., 1998. Regularized local linear prediction of chaotic time series. Physica D 112, 344–360.
Küsters, U., McCullough, B.D., Bell, M., 2006. Forecasting software: Past, present and future. Int. J. Forecast. 22, 599–615.
Law, A.M., 2015. Simulation Modeling and Analysis, fifth ed. McGrawHill Education, New York, NY.
Lee, Y.S., 2014. A semi-parametric approach for estimating critical fractiles under autocorrelated demand. European J. Oper. Res. 234, 163–173.
Lee, H.L., Padmanabhan, V., Whang, S., 1997. The bullwhip effect in supply chains. Sloan Manage. Rev. 38, 93–102.
Lieber, D., Stolpe, M., Konrad, B., Deuse, J., Morik, K., 2013. Quality prediction in interlinked manufacturing processes based on supervised & unsupervised machine learning. Procedia CIRP 7, 193–198.
Lisi, F., Medio, A., 1997. Is a random walk the best exchange rate predictor? Int. J. Forecast. 13, 255–267.
Lisi, F., Schiavo, R.A., 1999. A comparison between neural networks and chaotic models for exchange rate prediction. Comput. Statist. Data Anal. 30, 87–102.
Lora, A.T., Santos, J.M.R., Exposito, A.G., Ramos, J.L.M., Santos, J.C.R., 2007. Electricity market price forecasting based on weighted nearest neighbors techniques. IEEE Trans. Power Syst. 22, 1294–1301.
Lu, C.-J., 2014. Sales forecasting of computer products based on variable selection scheme and support vector regression. Neurocomputing 128, 491–499.
Makridakis, S., Hibon, M., 2000. The M3-competition: Results, conclusions and implications. Int. J. Forecast. 16, 451–476.
Manary, M.P., Willems, S.P., Shihata, A.F., 2009. Correcting heterogeneous and biased forecast error at intel for supply chain optimization. Interfaces 39, 415–427.
McNames, J., 1998. A nearest trajectory strategy for time prediction. In: Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling. pp. 112–128.
McNames, J., 1999. Innovations in Local Modeling for Time Series Prediction (Ph.D. thesis). Stanford University.
McNames, J., 2002. Local averaging optimization for chaotic time series prediction. Neurocomputing 48, 279–297.
Meade, N., 2002. A comparison of the accuracy of short term foreign exchange forecasting methods. Int. J. Forecast. 18, 67–83.
Minner, S., 2003. Multiple-supplier inventory models in supply chain management: A review. Int. J. Prod. Econ. 81–82, 265–279.
Mizrach, B., 1992. Multivariate nearest-neighbour forecasts of EMS exchange rates. J. Appl. Econometrics 7, S151–S163.
Montavon, G., Orr, G.B., Müller, K.-R. (Eds.), 2012. Neural Networks: Tricks of the Trade, second ed. In: Springer Lecture Notes in Computer Sciences, Springer-Verlag, Berlin, Heidelberg.
Mulhern, F.J., Caprara, R.J., 1994. A nearest neighbor model for forecasting market response. Int. J. Forecast. 10, 191–207.
Nemenyi, P., 1962. Distribution-free multiple comparisons. Biometrics 18, 263.
Nikolopoulos, K.I., Babai, M.Z., Bozos, K., 2016. Forecasting supply chain sporadic demand with nearest neighbor approaches. Int. J. Prod. Econ. 177, 139–148.
Nikolopoulos, K., Goodwin, P., Patelis, A., Assimakopoulos, V., 2007. Forecasting with cue information: A comparison of multiple regression with alternative forecasting approaches. European J. Oper. Res. 180, 354–368.
Papakostas, N., Efthymiou, K., Mourtzis, D., Chryssolouris, G., 2009. Modelling the complexity of manufacturing systems using nonlinear dynamics approaches. CIRP Ann. Manuf. Technol. 58, 437–440.
Peng, Y., Dong, M., Zuo, M.J., 2010. Current status of machine prognostics in condition-based maintenance: A review. Int. J. Adv. Manuf. Technol. 50, 297–313.
Petropoulos, F., Fildes, R., Goodwin, P., 2016. Do big losses in judgmental adjustments to statistical forecasts affect experts behaviour? European J. Oper. Res. 249, 842–852.
Petropoulos, F., Wang, X., Disney, S.M., 2019. The inventory performance of forecasting methods: Evidence from the M3 competition data. Int. J. Forecast. 35, 251–265.
Porter, M.E., 2008. The five competitive forces that shape strategy. Harv. Bus. Rev. 86, 78–94.
Prabhu, V.V., Duffie, N.A., 1995. Modelling and analysis of nonlinear dynamics in autonomous heterarchical manufacturing systems control. CIRP Ann. Manuf. Technol. 44, 425–428.
Prak, D., Teunter, R., Syntetos, A., 2017. On the calculation of safety stocks when demand is forecasted. European J. Oper. Res. 256, 454–461.

Ragwitz, M., Kantz, H., 2002. Markov Models from data by simple nonlinear time series predictors in delay embedding spaces. Phys. Rev. E (3) 65, 1–12.
Ramadge, P.J., 1993. Periodicity and chaos from switched flow systems: Contrasting examples of discretely controlled continuous systems. IEEE Trans. Automat. Control 38, 70–83.
Sauer, T., 1994. Time series prediction by using delay coordinate embedding. In: Weigend, A.S., Gershenfeld, N.A. (Eds.), Time Series Prediction: Forecasting the Future and Understanding the Past. Addison Wesley, Harlow, UK, pp. 175–193.
Sauer, T., Yorke, J.a., Casdagli, M., 1991. Embedology. J. Stat. Phys. 65, 579–616.
Sbrana, G., Silvestrini, A., 2014. Random switching exponential smoothing and inventory forecasting. Int. J. Prod. Econ. 156, 283–294.
Sbrana, G., Silvestrini, A., 2019. Random switching exponential smoothing: A new estimation approach. Int. J. Prod. Econ. 211, 211–220.
Scholz-Reiter, B., Freitag, M., Schmieder, A., 2002. Modelling and control of production systems based on nonlinear dynamics theory. CIRP Ann. Manuf. Technol. 51, 375–378.
Scholz-Reiter, B., Kück, M., Lappe, D., 2014. Prediction of customer demands for production planning - Automated selection and configuration of suitable prediction methods. CIRP Ann. Manuf. Technol. 63, 417–420.
Scholz-Reiter, B., Tervo, J.T., Freitag, M., 2006. Phase-synchronisation in continuous flow models of production networks. Physica A 363, 32–38.
She, D., Yang, X., 2010. A new adaptive local linear prediction method and its application in hydrological time series. Math. Probl. Eng. 2010, 1–15.
Sikorska, J.Z., Hodkiewicz, M., Ma, L., 2011. Prognostic modelling options for remaining useful life estimation by industry. Mech. Syst. Signal Process. 25, 1803–1836.
Silver, E.R., Pyke, D.F., Thomas, D.J., 2016. Inventory and Production Management in Supply Chains, Fourth Edition. Taylor & Francis.
Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. In: CRC Monographs on Statistics & Applied Probability, Chapman & Hall, London.
Singh, A., Damir, B., Deep, K., Ganju, A., 2015. Calibration of nearest neighbors model for Avalanche forecasting. Cold Reg. Sci. & Technol. 109, 33–42.
Stone, C.J., 1977. Consistent nonparametric regression. Ann. Statist. 5, 595–645.
Sugihara, G., May, R.M., 1990. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. Nature 344, 734–741.
Sun, H., Liu, H.X., Xiao, H., He, R.R., Ran, B., 2003. Use of local linear regression model for short-term traffic forecasting. Transp. Res. Rec. J. Transp. Res. Board 1836, 143–150.
Surana, A., Kumara, S., Greaves, M., Raghavan, U.N., 2005. Supply-chain networks: a complex adaptive systems perspective. Int. J. Prod. Res. 43, 4235–4265.
Svetunkov, I., Boylan, J.E., 2019. State-space ARIMA for supply-chain forecasting. Int. J. Prod. Res. 1–10.
Syntetos, A.A., Boylan, J.E., 2001. On the bias of intermittent demand estimates. Int. J. Prod. Econ. 71, 457–466.
Takens, F., 1981. Detecting strange attractors in turbulence. In: Dynamical Systems and Turbulence, Warwick 1980, vol. 898, pp. 366–381.
Takens, F., 2010 Reconstruction theory and nonlinear time series analysis. In: Handbook of Dynamical Systems, volume 3. Elsevier B.V., pp. 345–377, (Chapter 7).
Tashman, L.J., 2000. Out-of-sample tests of forecasting accuracy: an analysis and review. Int. J. Forecast. 16, 437–450.
Tempelmeier, H., 2008. Material-Logistik: Modelle Und Algorithmen Für Die Produktionsplanung Und -Steuerung in Advanced Planning-Systemen, seventh ed. Springer.
The MathWorks Inc., 2017. MATLAB Release 2017b. Natick, Massachusetts, United States.
Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B Stat. Methodol. 58, 267–288.
Trapero, J.R., Cardós, M., Kourentzes, N., 2019. Empirical safety stock estimation based on kernel and GARCH models. Omega 84, 199–211.
Van Wingerden, E., Basten, R.J.I., Dekker, R., Rustenburg, W.D., 2014. More grip on inventory control through improved forecasting: A comparative study at three companies. Int. J. Prod. Econ. 157, 220–237.
Wallström, P., Segerstedt, A., 2010. Evaluation of forecasting error measurements and techniques for intermittent demand. Int. J. Prod. Econ. 128, 625–636.
Wang, X., Petropoulos, F., 2016. To select or to combine? The inventory performance of model and expert forecasts. Int. J. Prod. Res. 54, 5271–5282.
Weigend, A.S., Gershenfeld, N.A. (Eds.), 1992. Time series prediction: Forecasting the future and understanding the past. In: Santa Fe Institute Studies in the Sciences of Complexity, Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis, Held in Santa Fe, New Mexico, May 14-17, 1992. Addison-Wesley, Reading, MA.
Weinert, N., Chiotellis, S., Seliger, G., 2011. Methodology for planning and operating energy-efficient production systems. CIRP Ann. Manuf. Technol. 60, 41–44.
Weron, R., 2014. Electricity price forecasting: A review of the state-of-the-art with a look into the future. Int. J. Forecast. 30, 1030–1081.
Wichard, J.D., 2011. Forecasting the NN5 time series with hybrid models. Int. J. Forecast. 27, 700–707.
Wiendahl, H.P., ElMaraghy, H.A., Nyhuis, P., Zäh, M.F., Wiendahl, H.H., Duffie, N., Brieke, M., 2007. Changeable manufacturing - classification, design and operation. CIRP Ann. Manuf. Technol. 56, 783–809.
Wiendahl, H.P., Scheffczyk, H., 1999. Simulation based analysis of complex production systems with methods of nonlinear dynamics. CIRP Ann. Manuf. Technol. 48, 357–360.
Wilding, R.D., 1998. Chaos theory: Implications for supply chain management. Int. J. Phys. Distrib. Logist. Manage. 9, 43–56.
Yakowitz, S., 1987. Nearest-neighbour methods for time series analysis. J. Time Series Anal. 8, 235–247.
Yankov, D., DeCoste, D., Keogh, E., 2006. Ensembles of nearest neighbor forecasts. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (Eds.), Machine Learning: ECML 2006. ECML 2006. Lecture Notes in Computer Science, vol 4212. Springer, Berlin, Heidelberg, pp. 545–556.