ARTICLE IN PRESS

International Journal of Forecasting xxx (xxxx) xxx

Contents lists available at ScienceDirect

International Journal of Forecasting

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

A machine learning-based framework for forecasting sales of new products with short life cycles using deep neural networks

Yara Kayyali Elalem [a,*], Sebastian Maier [b,a], Ralf W. Seifert [a,c]

[a] College of Management of Technology, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[b] Department of Statistical Science, University College London (UCL), United Kingdom
[c] International Institute for Management Development (IMD), Lausanne, Switzerland

ARTICLE INFO

Keywords:
Forecasting
Machine learning
Product life cycle
Analytics
Deep learning

ABSTRACT

Demand forecasting is becoming increasingly important as firms launch new products with short life cycles more frequently. This paper provides a framework based on state-of-the-art techniques that enables firms to use quantitative methods to forecast sales of newly launched, short-lived products that are similar to previous products when there is limited availability of historical sales data for the new product. In addition to exploiting historical data using time-series clustering, we perform data augmentation to generate sufficient sales data and consider two quantitative cluster assignment methods. We apply one traditional statistical (ARIMAX) and three machine learning methods based on deep neural networks (DNNs) – long short-term memory, gated recurrent units, and convolutional neural networks. Using two large data sets, we investigate the forecasting methods' comparative performance and, for the larger data set, show that clustering generally results in substantially lower forecast errors. Our key empirical finding is that simple ARIMAX considerably outperforms the more advanced DNNs, with mean absolute errors up to 21%–24% lower. However, when adding Gaussian white noise in our robustness analysis, we find that ARIMAX's performance deteriorates dramatically, whereas the considered DNNs display robust performance. Our results provide insights for practitioners on when to use advanced deep learning methods and when to use traditional methods.

© 2022 The Author(s). Published by Elsevier B.V. on behalf of International Institute of Forecasters. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Demand forecasting is a crucial element of supply chain management and is becoming increasingly important as firms bring new products with short life cycles to the market more frequently. While forecasting demand for existing products has improved steadily over the past decades, forecasting sales of newly launched, short-lived products remains challenging, due to the lack of abundant historical sales data (van Steenbergen & Mes, 2020). Furthermore, with shorter product life cycles (PLCs), the importance of accurate forecasting early on after a product is launched increases substantially (Basallo-Triana, Rodríguez-Sarasty, & Benitez-Restrepo, 2017). According to a cross-industry survey (Cooper & Edgett, 2012), new products contribute to an average of 27% of firms' revenues, yet profits from these products lag behind revenues. This is due to the high costs associated with product introduction, partly caused by the difficulty in forecasting sales of new products compared with more

* Corresponding author.
E-mail addresses: yara.kayyalielalem@epfl.ch (Y.K. Elalem), s.maier@ucl.ac.uk (S. Maier), ralf.seifert@epfl.ch (R.W. Seifert).

https://doi.org/10.1016/j.ijforecast.2022.09.005
0169-2070/© 2022 The Author(s). Published by Elsevier B.V. on behalf of International Institute of Forecasters. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Please cite this article as: Y.K. Elalem, S. Maier and R.W. Seifert, A machine learning-based framework for forecasting sales of new products with short life cycles using deep neural networks. International Journal of Forecasting (2022), https://doi.org/10.1016/j.ijforecast.2022.09.005.

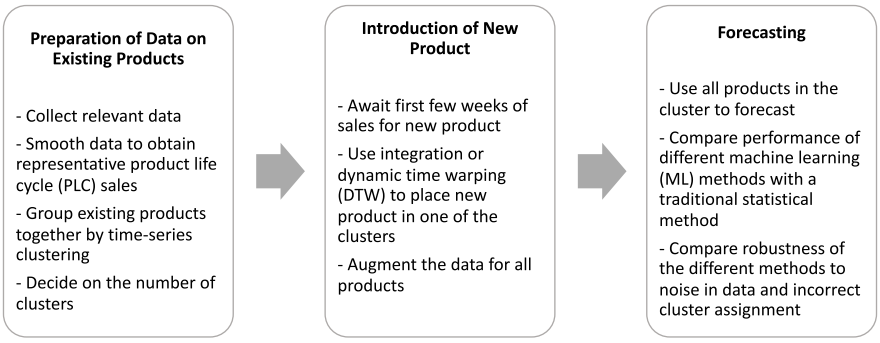


Fig. 1. Main steps of proposed framework for forecasting sales of new products with short life cycles.

stable ones that have been selling regularly in the market (Cecere, 2013). To increase profits from new products, it is crucial to generate more accurate post-launch forecasts.

Several studies show that sales of existing products are forecasted using statistical methods (Fildes & Goodwin, 2007). The key challenge for new products is the limited availability of historical data, thus preventing the use of more traditional statistical forecasting methods (Burruss & Kuettner, 2002). The dominant forecasting methods for new products are therefore market research, managerial opinions, and sales force input (Kahn, 2002). Although these techniques may be viable for forecasting sales of a new product, quantitative methods have the potential to substantially outperform these methods based on judgment (Sanders & Manrodt, 2003). Here, we seek to exploit the untapped potential of quantitative methods by deploying a range of state-of-the-art techniques in a novel way to forecast new product demand in data-scarce situations.

We develop a framework based on existing techniques that enables practitioners to apply quantitative methods to forecast sales of new products with short life cycles that are similar to previous products. An overview of our proposed framework is shown in Fig. 1. The framework builds on an approach widely used in research and industry: identify older, similar products to the new product, average their historical sales, and finally use the average sales as a base forecast (Baardman, Levin, Perakis, & Singhvi, 2018). The three main steps involved are: (1) prepare the sales data for existing products by smoothing the sales over their life cycle to obtain representative PLC sales, and then group similar products by means of clustering; (2) assign the new product to one of the clusters of existing products based on the first few weeks of new product sales using one of the two considered quantitative methods – integration and dynamic time warping (DTW); (3) perform data augmentation on all smoothed existing product sales in the cluster chosen and on the smoothed first few weeks of sales of the new product, then use the data to forecast sales over the rest of the new product's life cycle, applying both statistical and machine learning (ML) methods, as well as PLC shape-based methods, and, finally, compare the results under different conditions, including an analysis of the robustness of the quantitative methods to both white noise and an incorrect cluster assignment.

At this point it should be noted that while one sales data point of a new product would already be enough to apply the proposed forecasting framework, the more sales data available, the higher in general the forecasting accuracy. At the same time, it is also important to have older, similar products for the algorithms to train on. It should also be noted that our framework probably works best on products that sell for at least a couple of months or years before they are replaced by newer technology, such as those found in the electronics industry, including, but not limited to, phones, laptops, cameras, headphones, and speakers. Both data sets used in our study are those of personal computers. Other examples of suitable product categories include fashion products (not fast fashion), books, and movies. Although it could in principle be applied to new products with no similar old products or to fast fashion products that have only a few weeks of sales in the market, our developed forecasting framework should be expected to be less accurate in these situations.

This paper makes several contributions to the literature and practice. We propose a fully quantitative framework for forecasting sales of newly launched, short-lived products that are similar to previous products. More specifically, we describe two quantitative methods – integration and DTW – to position new products with short life cycles in clusters of similar products, rather than relying only on managerial opinions widely used in practice. Furthermore, we demonstrate how data augmentation, a common approach in ML, can be used to generate more data for quantitative forecasting methods. In particular, we use simple interpolation for data augmentation to properly define the statistical and ML models when used with limited data. We apply four different quantitative forecasting methods: one traditional statistical method – autoregressive integrated moving average with exogenous variables (ARIMAX) – and three advanced deep neural networks (DNNs) – long short-term memory (LSTM), gated recurrent units (GRUs), and convolutional neural networks (CNNs). In addition, we apply four PLC shape-based methods, including the well-known Bass model.

To operationalize the proposed framework and to evaluate the comparative performance of the forecasting methods considered, we use two distinct data sets: a publicly available Dell data set that comprises customer orders for 170 complete PLCs, and a second, much larger

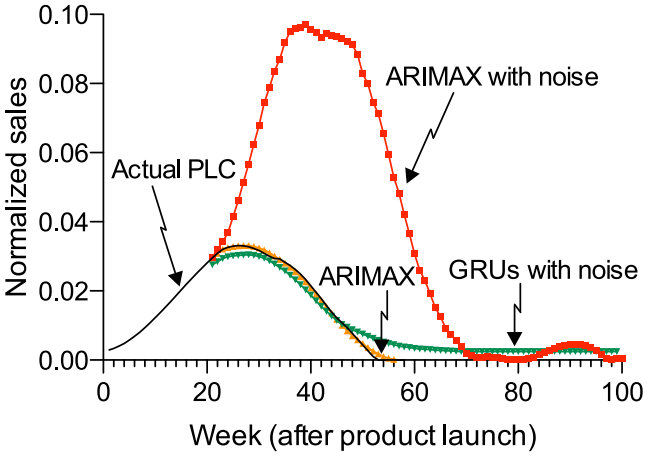


Fig. 2. Comparison between ARIMAX with and without noise (SD of 0.01) and GRUs with noise after 20 input weeks (Dell data).

data set – from Retailer X [1] – that includes the complete PLC order history for 843 products. The inputs of the quantitative forecasting methods are the first few weeks of sales of the product being forecasted and, depending on the type of analysis, the weekly time series that consist of chronological sales data (i.e. weekly customer orders) of all products in either a cluster or the entire data set. Lastly, we introduce additive white Gaussian noise as our basic noise model to test the robustness of the quantitative methods to noise in the data (see Fig. 2), and we intentionally assign new products to incorrect clusters to analyze the resulting impact on forecasting results.

This paper is organized as follows: Section 2 covers the relevant literature related to forecasting sales of new products. In Section 3 we describe the two data sets used in this work and the necessary data preparation. Section 4 presents our overall approach to time-series clustering, the two cluster assignment methods, and a comparison between them. In Section 5 we describe the data augmentation step, the different forecasting techniques used, and the computational implementation. Section 6 presents and discusses the forecasting results for the different methods. Section 7 analyzes the robustness of the quantitative methods under additive Gaussian white noise and under incorrect cluster assignments of products. Finally, in Section 8 we provide some concluding remarks.

2. Literature review

2.1. Product life cycle

The notion of a product life cycle (PLC) was introduced in 1950 in a *Harvard Business Review* article, "Pricing Policies for New Products", by Joel Dean. The main idea behind the PLC is that the invention of a new marketable product is first followed by a period where markets are still hesitant to try it, but the product is adopted over time and accepted in the market. Competitor encroachment then arises, and innovations narrow the gap of uniqueness between the product and its substitutes, turning the product into a commodity (Dean, 1976).

In the 1960s, the economist Raymond Vernon developed a more formal model known as Product Life Cycle Stages, or International Product Life Cycle (Vernon, 1966). According to Vernon, each product has a distinctive life cycle that starts with the product's development and ends with its decline. In general, PLC refers to primarily non-durable consumer goods and represents the sales curve from the time a product is introduced until it is removed from the market (Rink & Swan, 1979). It is usually pictured as a bell-shaped curve (similar to Fig. 2), and divided into several stages – introduction, growth, maturity, and decline – with each stage's lifetime ranging from a few weeks to several decades (Vernon, 1966).

Subsequently, many studies on different aspects of the PLC have been published. For example, Kurawarwala and Matsuo (1996) focused on forecasting and inventory management for short life-cycled products. Qi-zhi (2007) demonstrated the application of a diffusion model to increase forecasting accuracy for new products with short life cycles. Interestingly, as in the works cited, most of the research on products with short life cycles is illustrated by consumer electronics data, as they are representative of this class of products.

2.2. Quantitative models for forecasting sales of new products

Several growth models have been suggested for forecasting sales during the life cycle of a new product. The most commonly used product growth models are diffusion models. They are a set of stochastic modeling techniques that capture the life-cycle dynamics, estimate demand for new product categories, and direct important strategic choices in the pre-launch, launch, and post-launch phases (Kahn, 2014). One of the best-known diffusion models was developed by Bass (1969) to estimate how long it will take customers to start purchasing a new

[1] Retailer X is the disguised name of a real international electronics retailer. We use "X" instead of the retailer's actual name for confidentiality purposes.

product, considering that sales of the product grow to a peak after introduction, before decreasing and leveling off.

Since this early work, several reviews of extensions of diffusion models have been published. Meade and Islam (2006) discussed the importance of modifications of diffusion models to include exogenous variables related to both consumers and the market, and reviewed the works already done in the field. One of the future research directions they identified was forecasting new product diffusion with little or no data. Subsequently, Peres, Muller, and Mahajan (2010) reviewed works that focus on heterogeneity in customers' willingness to pay and different customer interactions that act as drivers to growth models.

One disadvantage of diffusion models is that, in general, they depend on historical data, which limits their usefulness for forecasting new technology sales (Lee & Lee, 2017). This is the case with the Bass model, where the parameters are usually estimated, which becomes difficult in situations where little is known about how the product will perform in the market. This limitation has resulted in many researchers attempting to improve the Bass diffusion model by making it more adaptive over time as more information becomes available (Lee & Lee, 2017).

In particular, Bayesian updating, which revises the parameters estimated by the Bass model once sufficient sales have been recorded in order to provide better forecasting results, has been integrated in several studies. For example, Zhu and Thonemann (2004) developed an adaptive forecasting algorithm of the Bass model. Their algorithm uses structural information about the PLC to model demand for products with short life cycles by combining the knowledge available before launch with the actual demand that becomes known when the product is released to the market. This enabled them to improve demand forecasts by continually updating the shape parameters of the Bass model.

2.3. Machine learning for forecasting sales of new products

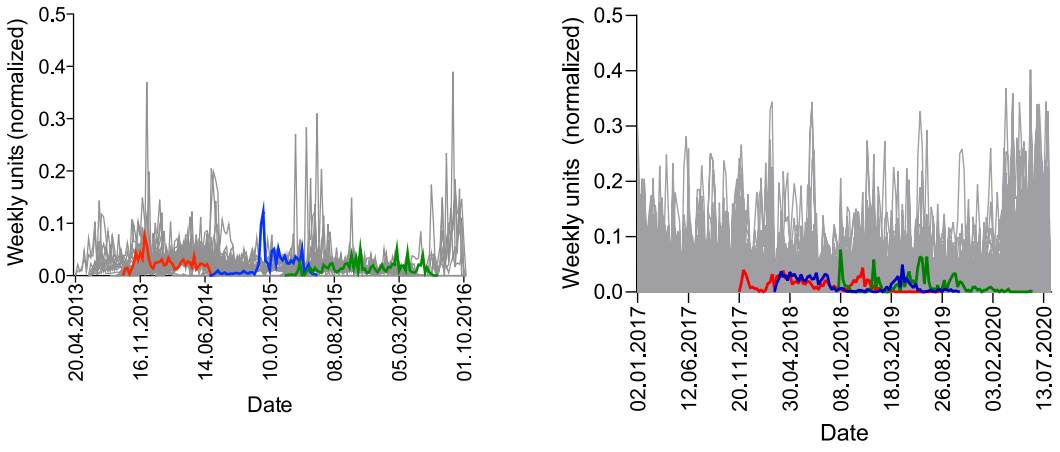
Mišić and Perakis (2019) reflected on how research in operations management has shifted from model-based approaches to data-driven analytical approaches that use data to create models, rather than applying known models to the data available. For example, Yildiz, Bilbao, and Sproul (2017) showed that artificial neural networks with Bayesian regulation backpropagation provide the highest forecasting accuracy when compared with other methods used for electricity load forecasting, which is an example of a field with sufficient data to apply such algorithms. Lu and Kao (2016) used K-means clustering to obtain different clusters of similar products. The authors related the sales data of the product they were forecasting to one of the clusters using different types of linkage methods, and then constructed an extreme learning machine model for that cluster. Their approach was applied to products with sufficient historical data and showed superior performance to forecasting without clustering. More recently, Petropoulos, Apiletti, Assimakopoulos, et al. (2022) provided a state-of-the-art overview of a wide range of forecasting methods and discussed advancements in new product forecasting.

The use of ML models in new product forecasting comes either through unsupervised learning by clustering or through supervised learning, which is applied after clustering. Techniques such as analogous forecasting may be used to overcome the lack of sufficient historical demand data (Meade & Islam, 2006), but often past similar products are grouped together by means of clustering and then the products in a cluster are used as a base for forecasting sales of a new product. For example, Hu, Acimovic, Erize, Thomas, and Van Mieghem (2018) used different types of clustering – feature-based, category-based, and data-driven – to group similar past products. Then, using managerial opinions, they placed a ready-to-launch new product in a particular cluster and applied traditional curve fitting to the average sales data in a cluster to forecast an entire PLC before product launch. By contrast, Thomassey and Happiette (2007) employed neural networks for clustering and classification. The authors obtained prototypes for each cluster, which is the mean life curve of all products in a cluster, and classified a new product using its descriptive characteristics. Then they used the prototype of that cluster as its sales forecast.

Other relevant works focused on clustering and then using ML methods for time-series forecasting. This allows the ML methods to be applied in a cross-learning manner, which means that they learn from multiple time series in a cluster to forecast one individual series (Makridakis, Spiliotis, & Assimakopoulos, 2020). For example, Fallah Tehrani and Ahrens (2016) used a probabilistic approach to classify fashion products depending on their sales, and then used a kernel machine approach to predict the units of sales of the new product. Basallo-Triana et al. (2017) proposed an analogue-based demand forecasting model for one-step forecasting by integrating multiple regression with fuzzy clustering. More recently, van Steenbergen and Mes (2020) developed a hybrid method combining K-means clustering, random forests, and quantile regression forests to forecast demand for new products (within 18 weeks of introduction) prior to product launch.

The research most closely related to ours is the recent work of Hu et al. (2018), who used different types of clustering and managerial opinions for cluster assignment. In this paper, we perform a similar clustering step to prepare the sales data for quantitative forecasting methods. However, instead of relying on managerial opinions, which are widely used in practice, we apply two quantitative methods – integration and DTW – and assess their performance in positioning a new product in its respective cluster. We then perform data augmentation, a common approach used in ML to generate artificial data points to reinforce the training of ML algorithms (DeVries & Taylor, 2017). To the best of our knowledge, we are the first to apply data augmentation to enhance the limited historical data of newly launched products in order to enable the use of quantitative forecasting methods.

Although Hu et al. (2018)'s study used clustering, which is an ML algorithm, they did not proceed to forecast sales of the new products using ML methods. They applied traditional curve fitting to forecast an entire PLC



(a) Dell data set. (b) Retailer X data set.

Fig. 3. Time series of the two data sets used in this study, with the red, blue, and green lines representing weekly normalized sales (before smoothing) for three randomly selected SKUs of each data set. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

before the product launch. In this work, we use four different quantitative methods – one traditional statistical and three DNN methods – to forecast sales after the new product's launch date (i.e. after the first demand is realized). Complementing their approach, we evaluate the comparative performance of the different methods under different conditions. Unlike (Hu et al., 2018), we report actual errors here, rather than relative errors based on proprietary forecasts, making our work directly replicable and, as such, comparable with future studies.

Our work is also somewhat related to Szozda (2010), who proposed a forecasting method for newly launched products with initial post-launch sales data, similar to the situation in our study. They compared the newly launched product sales to the time series of older products. However, the authors did not apply ML methods for sales forecasting, but used the time series of the closest older product by adjusting its demand patterns in both scale and length.

3. Data

3.1. Data sets

We use two distinct data sets for this study: the first is from Dell, and the second is from Retailer X (anonymized name). Dell is the third-largest producer of personal computers globally (Hamilton & Webster, 2012). The Dell data set is publicly available and consists of make-to-stock (MTS) products spanning multiple product categories such as fixed workstations, laptops, and desktops. Importantly, this data set was also used by Hu et al. (2018), and Acimovic, Erize, Hu, Thomas, and Van Mieghem (2018) provide further detail by describing the preprocessing steps taken to clean the data and render it ready to use.

The data consist of weekly customer orders for 170 stock-keeping units (SKUs) of personal computers that completed their life cycle from 2013 to 2016 (Acimovic et al., 2018). All the data are normalized and associated with the North American market. For our analysis, we use the data after filtering out the cancellations and configure-to-orders (CTOs). However, we consider all sales without truncating the end of the life cycle, as we are interested in how the product life cycles evolve even with external forces arising from managerial decisions such as promotions (refer to Acimovic et al., 2018 for more details). This allows us to forecast the life cycle of a product when it is in a market environment, where the PLC shape is generally affected by such factors.

Furthermore, all sales are normalized to a lifetime cumulative value of 1 by dividing the weekly customer orders per product by its total lifetime sales volume. This means that, for example, a value of 0.1 indicates that 10% of the total sales of a product occurred during that week, and a value of 0.5 indicates that 50% of the total sales of a product occurred in that week. Normalizing sales is crucial, since it defines the pattern of sales regardless of the actual quantities, thus making the models used less sensitive to different sales magnitudes. The volumes can then be adjusted depending on how the sales evolve. Several ML models train more efficiently in the presence of normalized data when the data have different ranges (Singh & Singh, 2020). Fig. 3(a) displays the full time series of all the 170 Dell products , i.e. the normalized weekly sales for each Dell product, along the selling horizon available and highlights three different PLCs selling in the beginning (in red), middle (in blue), and towards the end of the horizon (in green).

The second data set is from Retailer X, an international electronics retailer with both physical stores and online sales. The data set consists of normalized weekly sales data of 843 SKUs of personal computers that completed their life cycle from 2017 to mid-2020 in one market. The life cycle of a product in the data set starts when the first purchase of the product occurs and ends when no more purchases are made. The computers comprise different brands, such as Acer, Asus, Apple, HP, and Dell, with

Fig. 4. PLC curve fit by third-degree polynomial (SKU150, Dell data).

different models of the same brand differing by features such as processors and storage capacity. The data are pre-processed in a similar way to the Dell data set and are also normalized to 1 unit of life-cycle sales per SKU. The range of products in the Retailer X data set includes fixed workstations, laptops, and gaming computers, making the product portfolio similar to that of Dell.

The main differences between the two data sets is that Retailer X has almost five times as many SKUs as Dell and it comprises different brands of computers, including Dell. The PLC lengths in the Retailer X data set differ greatly compared to the Dell data set, and the PLCs of Retailer X do not necessarily follow the classical PLC shape, as some of the products have two maturity stages instead of just one, as exemplarily highlighted by the blue PLC curve in Fig. 3(b), which shows the full time series of all the 843 Retailer X products. Although most of the methodology and results shown in our work are for the Dell data, clustering and forecasting are also carried out on Retailer X's data to further demonstrate the consistency of the results obtained by our proposed framework and to strengthen the comparative performance evaluation.

3.2. Data preparation

Before clustering the products, it is necessary to have them all aligned at the same start date and to smooth the sales in order to remove displacement effects and noise from the data, respectively. All products are first shifted to a start date of "0," and then the normalized weekly sales for each product are smoothed to obtain the underlying representative PLC, thereby achieving better results for clustering and forecasting. Using normalized sales is important, since it groups products with similar PLCs together according to PLC shape, rather than sales volumes. Although it can be argued that shifting all PLCs to the same start date ignores the effect of seasonality, we are dealing with short product life cycles lasting for a few months to a few years in the market, so the effect of seasonality may not necessarily be apparent. It should be noted that seasonality effects and trends in the market can be accounted for by adjusting sales projections according to the launch data of the new product.

In order to smooth the PLCs, we applied a smoothing filter that fits polynomials throughout the data points for each time series. Note that negative values returned by the smoothing filter are set equal to zero because using negative values for smoothed sales would indicate that we account for phenomena such as customer returns and it would lead to negative training data (for the forecasting models), which is outside the scope of this paper. We chose a polynomial of the third degree, as it results in each PLC shape being close to a normal curve (see Fig. 4). This is being supported by a large body of research on the rate of sales of new products with short life cycles which has found that PLC shapes tend to follow normal distributions, with a peak at 50% penetration (for example, see Rogers, 1962; Mahajan, Muller, & Bass, 1990; Golder & Tellis, 2004).

4. Clustering

In this section, we cluster the PLCs of all existing products into different groups depending on similarities in sales patterns and life-cycle lengths. We then assign the newly launched product to a group of similar existing products that have already been sold in the market and use the sales data of that group's products as input for the quantitative forecasting techniques. In contrast to the approach of Hu et al. (2018), we do not normalize the PLC selling times. This is important as it allows us to forecast more accurately the expected time a product sells in the market, which is critical in order to decide, for example, when to release new products.

In our study, we apply data-driven time-series clustering. This type of clustering is capable of identifying hidden product attributes possibly unknown to demand planners and not represented in the raw data (Hu et al., 2018). More specifically, we deploy agglomerative hierarchical clustering. The main reason for this choice of clustering algorithm is that hierarchical clustering not only forms groups of similar products but also provides a graphical representation of the data, thereby making the choice of the number of clusters easier and more intuitive for practitioners (Özkoç, 2020) (see online Appendix A for a

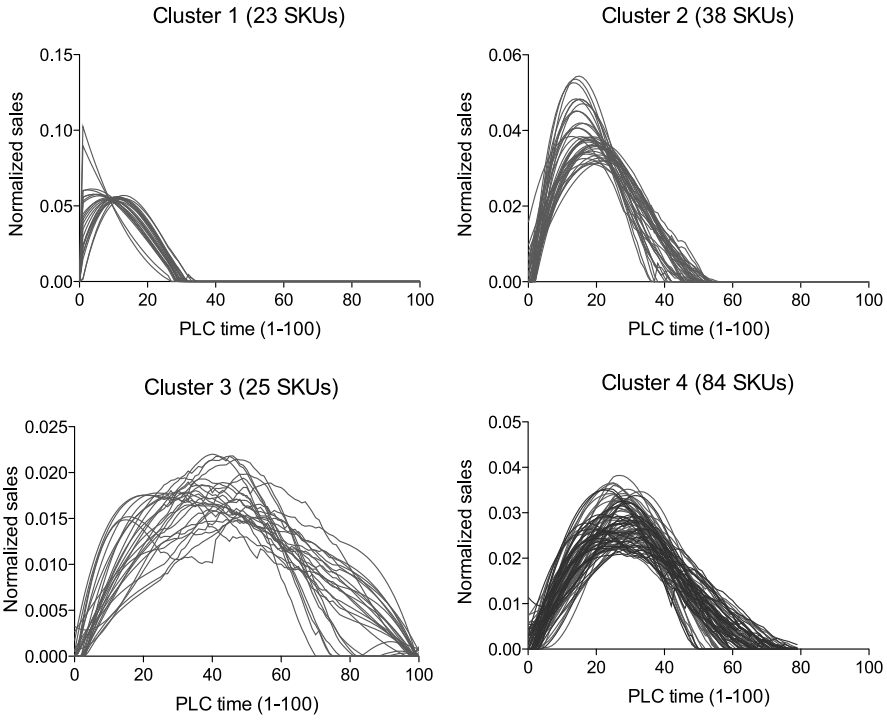


Fig. 5. Clustering results with individual curves both normalized and smoothed (Dell data).

graphical representation of results). Moreover, Javed, Lee, and Rizzo (2020) recently presented a time-series clustering benchmark based on 112 data sets using eight popular clustering methods, including agglomerative hierarchical clustering, and found that no method outperformed the others for all data sets. We have also tested several linkage methods: *single*, *complete*, *average*, and *ward*. Since the latter gave the best clustering results, we chose the *ward* linkage method for clustering. The results of the different linkage methods are presented in Appendix A in the supplementary material.

4.1. Overall approach

Our overall approach for clustering is as follows:

1. Shift all time series to the same starting date "0"; otherwise, the clustering step groups the products according to their start dates.
2. Smooth all individual PLCs so that the clustering step groups the products according to the underlying PLC shape and is not affected by weekly sales fluctuations.
3. Apply hierarchical clustering, a data-driven time-series clustering technique.
4. Choose the number of clusters by referring to the dendrogram generated from the clustering step (or from the scree plot).

The result of the clustering step is a dendrogram (see online Appendix A) that separates the products into different clusters using the Euclidean distance between each pair of PLCs. The variables used to calculate the Euclidean distance are the smoothed sales points of each time series. For each pair of products, i and j, the Euclidean distance between their smoothed PLCs, $ED(\tilde{D}^{a,i}, \tilde{D}^{a,j})$, is calculated as follows:

$$ED(\tilde{D}^{a,i}, \tilde{D}^{a,j}) = \sqrt{\sum_{t=0}^{n} \left(\tilde{D}_t^{a,i} - \tilde{D}_t^{a,j}\right)^2}, \quad (1)$$

where $(\tilde{D}_t^{a,i}, \tilde{D}_t^{a,j})$ is the pair of smoothed sales at time t, and n is the entire length of the time series.

Note that the Euclidean distance biases the clustering results when PLCs are not aligned to the same starting date. This is because products are clustered predominantly based on their launch dates rather than similarities in sales patterns. So shifting products to the same starting date allows similarities in sales patterns to be the dominant clustering feature. This is important for analogous forecasting, which uses sales information about similar past products to forecast sales of a new product. However, when the time series are all shifted to the same starting date, the forecaster must pay attention to any seasonality and trend effects of the old time series and then consider adding similar effects to the newly generated forecasts depending on the selling season.

As the threshold distance between the clusters increases, the number of clusters needed to separate the SKUs decreases, until all products are eventually grouped into one big cluster. The number of clusters can be determined either visually from the dendrogram by determining a horizontal line that cuts through the number of clusters needed, or by producing a scree plot. These curves illustrate the percentage variance in each cluster

and how this variance decreases as the number of clusters increases. We chose four clusters, since this results in a meaningful separation in the dendrogram and explains almost 80% of the variance within each cluster of the Dell data set. Fig. 5 shows the result of time-series clustering for the Dell data set; the four clusters are distinguished by different PLC behavior, peak sales amounts (note the difference in the *y*-axis scale), and life-cycle lengths. We also chose four clusters for the Retailer X data set and report all the clustering steps in online Appendix A, which reveals not only the larger number of SKUs but also the higher variance in the time-series curves of our second data set. In Appendix B of the supplementary material, we illustrate the results of correlation heat maps between SKUs in each cluster for both data sets to show the similarity between the different PLC curves.

It is important to note that clustering was first tested without the smoothing step. However, the results of the dendrogram were poor, as the products were not separated according to their underlying sales patterns. Therefore, the smoothing step is deemed necessary, since it removes noise in the data, which may adversely affect clustering.

4.2. Assigning products to clusters

After establishing the clusters, we present two quantitative methods – integration and dynamic time warping (DTW) – to assign a new product to one of the four clusters within a few weeks of its launch. In industry, managers typically relate new products to old ones using their judgment (Hu et al., 2018). By contrast, we apply two quantitative methods that do not rely on managers' judgments but, in practice, can be used to support managers' opinions and verify their choice.

To test the effectiveness of the two methods at assigning a new product to the correct cluster, we first remove the product we want to test completely from the data set. With the remaining curves already assigned to one of the four clusters, and knowing which cluster the "new" product belonged to before removing it, we can proceed to evaluate the two methods. Specifically, we apply the following five-step approach:

1. Generate an average representative curve for each cluster by calculating the average weekly sales of all products in a cluster at each time step. This substantially decreases the computational time needed, since the newly launched product is compared with four representative curves, rather than with all the remaining curves in the data set.
2. Wait for the first few weeks of sales of the new product.
3. Use integration or DTW to position the new product in one of the clusters, giving the mean absolute error (MAE) between the new product and each cluster.
4. Assign the new product to the cluster with the lowest MAE.
5. Repeat steps 1–4 for three different groups of 15% of the data set (a percentage commonly used for validation) for each assignment method, and report the average percentage of correct assignments per method for different weeks of sales after product launch. Starting from one week, we test up to 20 weeks, since some products, especially those in clusters 1 and 2, have entire life cycles of just 30 weeks.

For steps 3 and 4, we repeated the measurements using the mean squared error (MSE) and compared the error measures obtained to those found using the MAE measure but did not find notable differences. The reader is referred to online Appendices C and D for a description of the integration and DTW methods, respectively, and a comparison of them using the MSE can be found in online Appendix E. In the following section we compare their performance at correctly assigning new products to their respective clusters.

4.3. Comparison between integration and DTW

To compare the effectiveness of using integration versus DTW in assigning a new product to its correct cluster, we measure the percentage of correct assignments per method for different weeks after a new product is launched. Fig. 6 illustrates the percentage of correct assignments for both methods. The results show that integration follows a more gradual increase in correct assignments than DTW as more weeks of sales data of the new products are available. Because all the data have been shifted to the same start date, and because the PLCs follow a very similar pattern, albeit with different shape parameters, the power of DTW is somewhat limited for these data. As the number of weeks after product launch increases to five, DTW can find some differences between the clusters, but after that, the similarities become too high. As a result, DTW assigns the products randomly, giving the same average percentage of correct assignments even if more weeks are available. Interestingly, the small peak at seven weeks after product launch suggests that at this point in time, the difference in shape between the different clusters is detected; however, beyond this point the clusters continue behaving in a similar way, decreasing the assignment accuracy.

It is important to recall that for DTW, even if there are transformations in the amplitude or period of the curves, the algorithm considers them to be similar, unlike the integration method, which uses these transformations to detect differences in the rate of sales. We use the Euclidean distance as a similarity measure for clustering. Since the Euclidean distance measures the similarity between each pair of points at the same time step between two curves, curves will be grouped together if they have similar slopes. Integration in this case gives better results in assigning products to the correct clusters, as it gives a measurement of the rate of sales of the products, which is represented through the slope measures. As more weeks pass, the rate of sales of the new product (i.e. the criterion that integration relies on) becomes more similar to the representative cluster, and that is why integration

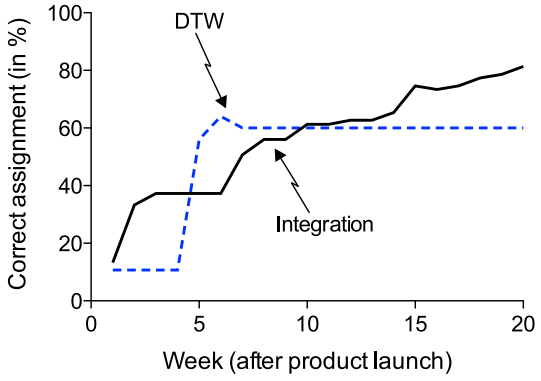


Fig. 6. Percentage of correct cluster assignment achieved by DTW and integration for various weeks after product launch (Dell data).

generally produces better results. For example, 15 weeks after the new product's launch, it is apparent from Fig. 5 that if the product is in the decline phase at that point, it belongs to cluster 1; if it is in the maturity phase, it belongs to cluster 2; and if it is in the growth phase, it belongs to either cluster 3 or 4. This also explains why there is a bump in the percentage of correct integration assignments at 15 weeks after product launch (Fig. 6), as the clusters are easily distinguished at that point in time.

When repeating the clustering using DTW as the distance measure instead of the Euclidean distance, and then assigning new products to clusters, we find that DTW outperforms integration in cluster assignment (see online Appendix F in the supplementary material). By using DTW as a distance measure, products are clustered based on similarities despite temporal shifts, rather than based on slopes and evolution of sales. Therefore, if two time series are similar in shape but have different slopes and sales evolution, DTW-based clustering groups them together, while integration-based clustering no longer provides good performance. As a result, cluster assignment using DTW will start to assign products more correctly. However, due to the nature of DTW, it may be argued that using it for only parts of curves (that is, few weeks of sales after introduction) may be unsuitable. Moreover, the dendrogram results from using the Euclidean distance as a measure show greater separation between the clusters than using DTW as a distance measure, as shown and discussed in online Appendix F.

5. Forecasting

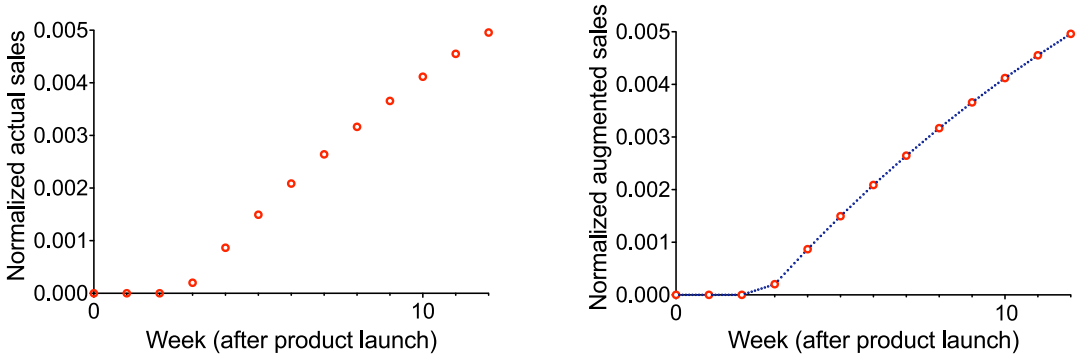
In this section, we first describe how we enhance the data to make them suitable for the use of quantitative methods, before testing and comparing a traditional statistical forecasting method with three types of deep learning ML methods for sales forecasting. See online Appendix G for details on the implementation and parameter estimation of the quantitative methods. We apply the methods using smoothed sales, with and without the clustering step. Doing so allows us to verify the need for the clustering step by evaluating its importance in the forecasting results. As a benchmark for the shape-free quantitative methods in our comparative performance analysis, we apply three forecasting methods that are based on fitting families of curves to historical PLC shapes. Importantly, for all the methods applied here, we assume at this point that we know to which cluster the new product belongs. The input of the quantitative forecasting methods is in the form of weekly time series (of chronological sales history) and consists of both the first few weeks of sales of the new product and the historical sales either of all products in the relevant cluster or, if clustering is not used, of all products in the entire data set.

5.1. Data augmentation

To ensure that the quantitative methods have sufficient training data, we generate more data using data augmentation (DeVries & Taylor, 2017). This step is necessary when the algorithms require abundant data points to perform well but the training data available are very limited, as in our case with newly introduced products with short life cycles. The PLCs in our data sets have between 30 and 100 data points, meaning that products in our data sets experience at most 100 weeks of sales. We generate nine additional points between each pair of sales by interpolation, before using the quantitative forecasting methods. We chose nine additional points because this increases the forecasting accuracy while resulting in an acceptable running time, as described in the following paragraph. These additional points fall on the curve of each product's sales, reinforcing the shape of the sales curve yet not affecting the weekly sales volumes. We perform data augmentation for all the curves input into the forecasting algorithms. However, after using these augmented points to support the training of the algorithms, we disregard them and consider only the actual points of sales when calculating the forecast errors.

By augmenting nine additional data points, the percentage of original data points to all data points on the augmented new time series – which consists of original plus augmented ones – is 10%. The choice behind the percentage of actual to augmented data points involves an important tradeoff: while data augmentation may be necessary to properly train ML algorithms and avoid over-fitting with limited data, too much use of data augmentation may lead to under-fitting, which is also undesirable (Park et al., 2019). To avoid under-fitting, the models would have to be more complex, which, however, increases computational costs. A ratio of 10% represents an appropriate tradeoff between forecasting accuracy and computational time.

Many of the data augmentation techniques for time series increase the number of time series for classification purposes. These include *jittering*, where white noise is added to some of the time series to generate new time series; *scaling*, where for a given time series the values are scaled by a given factor to generate new time series; *permutations*, where segments of the time series are rearranged to produce new time series; and *averaging and interpolation*, where a new time series is created from the combination of two time series and located between the original time series (see Iwana & Uchida, 2021



(a) Without augmentation (b) With augmentation

Fig. 7. Example of data augmentation results on Dell data set (first weeks of SKU 105).

for more data augmentation techniques for time series). However, since our data sets have sufficient time series for classification, but limited data points per time series for forecasting, we use a data augmentation technique that increases the amount of data points per time series, rather than the total number of time series available.

We note that the data augmentation step is done only for the quantitative methods and not for the PLC shape-based methods. The reason for this is that the latter rely on curve-fitting, meaning that they optimize fitting as many data points through curves of different parameters, until the optimal parameters are found. So if we used curve-fitting with data augmentation, we would be forcing the curves to fit through the augmented data points, giving fewer degrees of freedom to fit through the actual data points. Fig. 7 illustrates the results of data augmentation considering the first few weeks of sales of an SKU in the Dell data set, showing the actual smoothed sales data in red (left) and the same data points with the additional augmented data points in blue (right). After data augmentation, the curve is supported by more curve-defining data points and the shape of the PLC curve is better defined, which enhances the forecasting performance.

Importantly, we ensured that the augmented points are equidistant both from one another and from the original sales data when interpolating these additional data points. The reason behind positioning all (augmented and original) data points equidistant from one another is that the augmented data points and the original sales data act together as the new input to the forecasting algorithms. If the input data points were not equidistant from one another, it would translate into giving the algorithms data with different x-positions and therefore different frequencies (e.g. some daily and some monthly data points). This would alter the skewness of the curves, as the algorithms consider the time intervals between the points equidistant from one another, requiring the forecaster to reconcile the different frequencies into one before using the data for forecasting.

5.2. Traditional statistical model

We apply a version of the autoregressive integrated moving average (ARIMA) called ARIMAX as the traditional statistical forecasting method and as a baseline for comparison with the DNN methods. ARIMA is best suited to short-term forecasting for around 12 months ahead (Stellwagen & Tashman, 2013), making it suitable for forecasting sales of short-lived products. ARIMAX allows for the use of exogenous (or explanatory) variables (the "X" in ARIMAX) within the algorithm. Examples of exogenous variables are prices and weather conditions. In our case, the exogenous variables are the smoothed sales values of all (older) products in either a cluster or the entire data set.

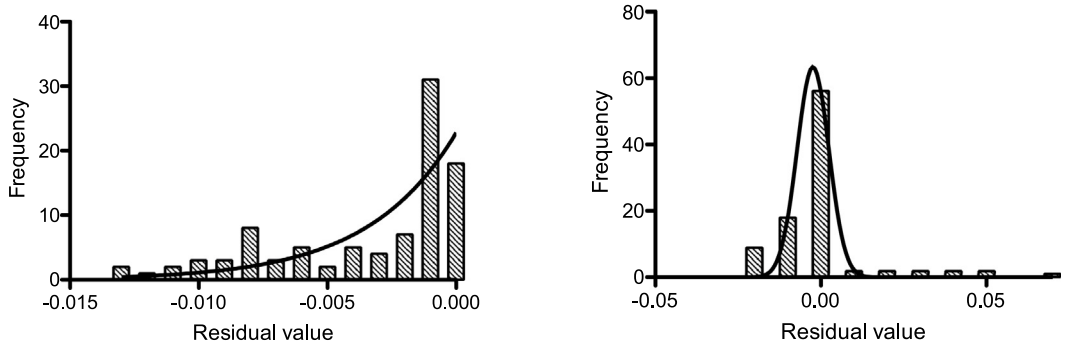
Extending the ARIMA model, the mathematical formulation of ARIMAX is given by

$$y_t^* = \sum_{i=1}^{n} \beta_i x_t^i + \mu + \sum_{i=1}^{p} \phi_i y_{t-i}^* + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t, \qquad (2)$$

where β_i is the coefficient associated with the smoothed sales of product i at time t, x_t^i; n denotes the number of older products or curves present in the cluster; μ denotes the intercept coefficient; $y_t^* = \Delta^d y_t$ denotes the observations of the target time-series values differenced d times to fulfill the stationary requirements; ϕ and θ are the coefficients of the AR and MA parts, respectively; and ϵ denotes the prediction error.

As a general rule of thumb, ARIMA typically requires 50–100 data points as input (Box & Tiao, 1975). In Section 6 we forecast with a minimum input of 10 and six introduction weeks for the Dell and Retailer X data sets, respectively. Therefore, adding nine data points between each pair of sales data points augments the 10 weeks of sales to 100 data points and the six weeks of sales to 60 data points, making the augmented data set suitable for use with ARIMAX.

To validate our choice of nine additional data points, we analyze the distribution of the residuals (errors) of the fitted ARIMAX models. If the ARIMAX model is correctly specified, then the residuals should be normally distributed with mean zero. In Fig. 8 we show an example of the distribution of the residuals considering SKU 10 of the Dell data set. The figure shows the difference between the distribution before and after the data augmentation step, and how adding nine data points changes the distribution of the errors, indicating that the augmented data set is appropriate for the model. In Table 1 we report



(a) Without augmentation (b) With augmentation

Fig. 8. Example of histogram of residuals without and with data augmentation (SKU 10, Dell data).

Table 1
Best-fit values for Gaussian distribution (SKU 10, Dell data).

Metric	Without augmentation	With augmentation
Mean	0.1761	−0.002387
Standard deviation	0.02453	0.004813
Amplitude	3.548e+012	63.29
R^2	n/a	0.9649

the best-fit values for fitting a normal distribution to the two histograms. It can be seen that fitting a Gaussian curve without data augmentation does not converge (no R^2), whereas applying data augmentation leads to a sufficiently high R^2 of 0.9646, which confirms that our choice of nine-fold augmented data sets is appropriate.

5.3. Machine learning models

For the ML methods, we apply three types of DNNs: long short-term memory (LSTM), gated recurrent units (GRUs), and convolutional neural networks (CNNs). The main reason for choosing DNNs is that they, like ARIMAX, can be multivariate, meaning that sales of other (e.g. old) products can be fed into the algorithms when forecasting sales of the new product. DNNs are an extension of single-layered artificial neural networks to multiple layers and have been tested and compared with other time-series forecasting methods in many studies, often giving the best results (see Cao, Ewing, and Thompson (2012), Zhang, Shen, Zhao, and Yang (2017), Paliari, Karanikola, and Kotsiantis (2021), and Hu, Wang, Ye, and Wang (2021)). We also tested XGBoost, but it did not provide meaningful forecasting results for the considered numbers of sales weeks of the new product, so we excluded it from our report. Recently, Kraus, Feuerriegel, and Oztekin (2020) discussed the importance of deep learning in the field of business analytics (including sales forecasting) and argued that DNNs are able to identify previously unknown, potentially useful patterns more accurately than other widely used predictive models such as support vector machines and random forests.

Recurrent neural networks (RNNs) such as LSTM and GRUs take sequential data as input and comprise a network $f_{\text{RNN}} : \mathbb{X} \to \mathbb{Y}$, where the sequences in $\mathbb{X}$ have the form of $\boldsymbol{x}^i = [x_1^i, x_2^i, \ldots x_\tau^i]$, with x^i representing the input variable, such as the sales time series of one old product or the new product, given τ input week(s). The network also has internal hidden layers, denoted by $h_1^i, \ldots, h_\tau^i$. The knowledge of the sequence is accumulated in these hidden states. For a given time $t \in \{1, \ldots, \tau\}$, the input to the network is the concatenation of x_t^i and the previous hidden state h_{t-1}^i. The output of the RNN is therefore a computation of the following:

$$f_{\text{RNN}}(x_1^i, \ldots, x_\tau^i) = f_{\text{DNN}}([x_\tau^i, f_{\text{DNN}}([x_{\tau-1}^i, \ldots, f_{\text{DNN}}([x_1^i]; W, b); \ldots]; W, b]; W, b), \quad (3)$$

where f_{DNN} is a deep layer of neural networks composed of k layers of single neural networks, given by

$$f_{\text{DNN}}(\boldsymbol{x}) = \underbrace{f_{\text{NN}}(f_{\text{NN}}(\ldots f_{\text{NN}}}_{\text{k}}(\boldsymbol{x}))). \quad (4)$$

Each neural network f_{NN} is computed via an activation function and a linear combination as follows:

$$f_{\text{NN}}(\boldsymbol{x}; W, b) = \sigma(W\boldsymbol{x} + b), \quad (5)$$

where σ is the activation function, which in our case is the rectified linear unit (ReLU) function given by $\sigma(x) = \max(0, x) \in [0, \infty)$; W is the weight matrix; and b is the intercept. We refer the reader to Kraus et al. (2020) for more information regarding the architecture of RNNs and CNNs and the optimization of model parameters.

Long short-term memory. LSTM is a type of RNN that addresses the problem of vanishing gradients faced by ordinary RNNs (Hochreiter & Schmidhuber, 1997). It does so by learning to bridge time intervals without the loss of short time-lag capabilities, allowing the network to *remember* both long- and short-term patterns in the data. The architecture consists of several gated units and memory cells to facilitate information storage over time, in addition to the other layers found in RNNs.

Gated recurrent units. GRUs are the second most common type of RNN behind LSTM. GRUs are similar to LSTM in that they operate through gates to overcome the vanishing gradient problem, but they differ in the type of gates used. GRUs use reset and update gates, whereas LSTM uses input, output, and forget gates. Many studies

Table 2
Summary of PLC shape-based methods.

Base curve family	Forecasted demand at time t	Parameters
Bass	$\hat{D}_t^{Bass} = \frac{p(p+q)^2 e^{-(p+q)^2}}{(p+qe^{-(p+q)t})^2}$	(p, q): shape parameters
Triangle	$\hat{D}_t^{triangle} = \begin{cases} at+b & 0<t<t_1 \\ c(t-t_1)+(at_1+b) & t_1 \le t \le T \end{cases}$	(a, b, c) : shape parameters t_1 : turning point T : life-cycle length
Trapezoid	$\hat{D}_t^{trapezoid} = \begin{cases} at+b & 0<t<t_1 \\ at_1+b & t_1 \le t < t_2 \\ c(t-t_2)+(at_1+b) & t_2 \le t \le T \end{cases}$	(a, b, c) : shape parameters t_1 : beginning of maturity stage t_2 : end of maturity stage T : life-cycle length
Poly$n$	$\hat{D}_t^{polyn} = \sum_{j=0}^{n} a_j t^j$	a_j : shape parameters for $j = 0, \ldots, n$

demonstrate the applications of GRUs, with (Kumar, Hussain, Banarjee, & Reza, 2018) showing that both LSTM and GRUs perform best when used for forecasting.

Convolutional neural networks. CNNs are a type of artificial neural network commonly used in image processing. The aim is to form spatial filters and convolve these filters over each channel in an image (Rudin & Carlson, 2019) before passing the input to the next layer. Therefore, the number of filters in a CNN should be optimized. Although their use mostly involves image processing, some studies have applied CNNs to forecast time series. For example, Liu, Hou, and Liu (2017) used a CNN-based model to forecast foreign exchange rates and showed that its performance for long-term forecasts outperforms other ML models, even when compared with GRUs. We therefore include it in our study.

5.4. PLC shape-based methods

To complement our analysis, we also apply three forecasting methods that are based on fitting families of curves – Bass diffusion, piece-wise linear, and polynomial curves – to PLC shapes. Using these families of curves was proposed by Hu et al. (2018), and their mathematical formulations are summarized in Table 2. We use the methods on our two data sets after the smoothing and clustering steps. As in the work of Hu et al. (2018), we apply two different approaches to fit the models' parameters: "*(a) Taking the average of similar curves (GenerateAvg)*" for each cluster; and "*(b) Fitting the best curve through the data points (GenerateFit)*" in each cluster. For *GenerateAvg*, we simply calculate the average weekly sales of all the products in a cluster at each time step (similar to Section 4.2). For *GenerateFit*, we generate a representative curve in a similar fashion to Hu et al. (2018) by formulating an optimization problem that, depending on the curve (bass, polynomial, triangle, and trapezoid), finds the parameter values that minimize the sum of squared errors across all products in the cluster over all time steps.

Note that for the Bass diffusion model, we denote the forecasting results (in Tables 3 and 4) obtained through the two approaches described above by Bass-I. We then use the Bass model in a different manner by applying it to only one PLC – the PLC of the most similar older product in the cluster – instead of applying it to the average PLC curve per cluster. More specifically, we first calculate the MSE between the data of the first few weeks of sales of the new product and the first few weeks of each PLC in the cluster. The PLC in the cluster with the lowest MSE value is assumed to have the highest similarity to the new product. After finding the most similar PLC of an old product, we fit the Bass parameters to that PLC and use it for forecasting. Denoted by Bass-II, this approach was inspired by Zhu and Thonemann (2004) in which, over time, new products are compared to old products and the parameters of the Bass model are updated accordingly. Importantly, by using Bass-II, the Bass model uses the new product's sales data available after a few weeks of its launch and compares the data to sales of older products, thus taking advantage of the sales information available at that point.

In Fig. 9, we provide a small example to demonstrate how the Bass-II approach works. Assume we place the "new product" after 20 weeks of launch in cluster 3, which has only two older products: "old product 1" and "old product 2." Considering the first 20 weeks, the MSE measure between the new product and old product 1 is lowest, so the sales of the new product are closest to old product 1. We therefore fit the Bass-II model parameters, p and q, to the sales time series of old product 1 and use those parameters to forecast the rest of the PLC of the new product.

It should be noted, however, that since the Bass-II model parameters are obtained based on estimates from a single old PLC, the results can be extremely poor if it turns out that the old PLC has a very different sales pattern afterwards. In our case, the old and new PLCs are very similar, so this approach can be used. But if the new product is expected to behave very differently from the old products, this approach should not be considered, as it relies on the sales of a single old product during forecasting and does not benefit from learning sales patterns from different PLCs.

We also tried for the Triangle and Trapezoid to use the additional information by fitting their a and b parameters to the first few weeks of sales data, but this gave poor results, since these few data points are, in general, insufficient to estimate these parameters accurately.

For each PLC shape-based method, we calculate the parameters for both *GenerateAvg* and *GenerateFit* twice: (*1*)

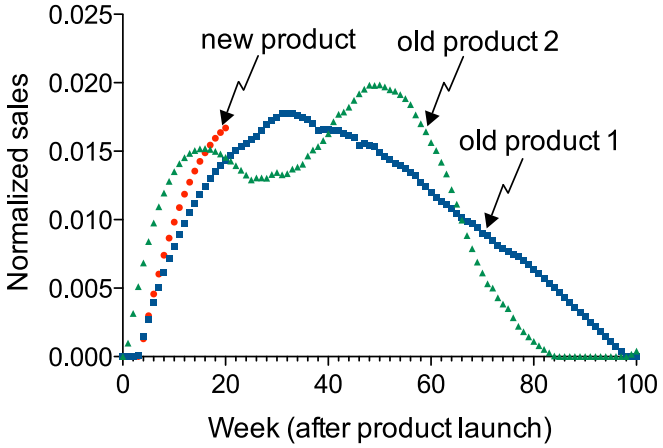


Fig. 9. Reference to how the Bass-II model works.

with respect to the *GenerateAvg* and *GenerateFit* of each cluster independent of the number of weeks of sales of the new product; and (*2*) with respect to the *GenerateAvg* and *GenerateFit* of each cluster starting from the number of weeks of sales already encountered until the end of the new product's life cycle. We apply the second strategy to make use of the additional sales information we have available – by truncating the beginning (up to the new product's sales weeks encountered) of the representative curves generated through the *GenerateAvg* and *GenerateFit* approaches – and report the results of the more accurate parameter calculation strategy. Superscripts *1* and *2* in Tables 3 and 4 denote which of the two strategies performed best, and the results for both strategies are reported in online Appendix H.

6. Forecasting results

This section reports the forecasting results on both the Dell and Retailer X data sets, using ARIMAX and the three considered DNNs – LSTM, GRUs, and CNNs – with and without clustering, considering different weeks after the introduction of the new product. It also includes a comparison of the forecast accuracy of the quantitative methods and the PLC shape-based methods. Note that the results in terms of forecast accuracy reported here are calculated between the forecasted sales and the actual (unsmoothed) sales of the product over its life cycle, across three groups of 15% of the data set, equal to 25 SKUs per group. To evaluate the out-of-sample forecast accuracy, we measure the mean absolute scaled error (MASE) and the sum of absolute errors (SAE). These two error measurements are widely and commonly used in the related literature. The errors are calculated per product between the forecasted and actual (unsmoothed) sales at each time step, and are then averaged over the group of SKUs tested, as described in online Appendix I. Note that for both data sets the parameters of ARIMAX and those of the two Bass models are statistically significant at the 5% level. In online Appendix J, we show the error distributions over the different SKUs for each forecasting method used.

6.1. Dell data set

We start with the Dell data set and evaluate the sales forecast accuracy over the new product's entire life cycle, at 10, 15, and 20 weeks after a new product is launched. We chose 20 weeks as the maximum number of weeks because products in clusters 1 and 2 would already be in their decline stage after that. The forecast errors in Table 3 show that ARIMAX achieved the highest forecast accuracy. When compared to the best-performing DNN – either LSTM or GRUs – the forecast errors in terms of the MASE (SAE) achieved by ARIMAX are 24.32% (22.86%), 23.64% (37.14%), and 21.10% (39.29%) lower for 10, 15, and 20 input weeks, respectively. On the other hand, when compared with the worst-performing DNN – CNNs – ARIMAX achieved even higher reductions of 38.69% (47.06%), 34.38% (46.34%), and 37.23% (50%), respectively. Even though ARIMAX without clustering gave the best results, LSTM and GRUs also performed relatively well. However, CNNs exhibited the poorest performance compared with the two RNN methods, in contrast to what was found by Liu et al. (2017) in the context of long-term forecasting of exchange rates.

It can be observed from Table 3 that the PLC shape-based methods produce relatively accurate forecasts and, on average, perform better than the DNNs. In fact, when ARIMAX is compared to the best-performing curve-fitting method – either poly2 (Avg) or Bass-II – the forecast errors in terms of the MASE (SAE) achieved by ARIMAX are 9.68% (25%), 10.64% (31.25%), and 11.34% (34.61%) lower for 10, 15, and 20 input weeks, respectively. This can be explained by the relatively small size of the Dell data set and by the PLCs of the included products, which follow the classical life-cycle pattern with well-defined PLC stages.

Fig. 10 displays the forecasting results of ARIMAX and GRUs for an example SKU with 10 weeks of post-launch sales data. Note that the forecasting is done before clustering for ARIMAX and after clustering for GRUs, as this gives the best results. It can be seen that the ARIMAX outperforms GRUs, because the forecasted sales points of ARIMAX lie closer to the actual sales curve than those of

Table 3
Forecast errors for new products for different weeks after introduction (Dell data).

Quantitative method	Time-series clustering	MASE			SAE		
		10 weeks	15 weeks	20 weeks	10 weeks	15 weeks	20 weeks
LSTM	No	1.14	1.13	1.18	0.43	0.36	0.30
	Yes	1.13	1.13	1.09	0.35	0.35	0.28
GRUs	No	1.16	1.10	1.10	0.44	0.35	0.32
	Yes	1.11	1.11	1.12	0.42	0.35	0.29
CNNs	No	1.37	1.23	1.20	0.51	0.38	0.30
	Yes	1.32	1.28	1.37	0.49	0.41	0.34
ARIMAX	No	0.84	0.84	0.86	0.27	0.22	0.17
	Yes	0.97	0.99	1.06	0.37	0.32	0.26
Base curve family	Cluster curve generation	MASE			SAE		
		10 weeks	15 weeks	20 weeks	10 weeks	15 weeks	20 weeks
Bass-I	Fit[1]	0.99	1.01	1.05	0.38	0.33	0.28
	Avg[1]	0.97	1.00	1.04	0.38	0.33	0.28
poly2	Fit[1]	1.07	1.10	1.18	0.41	0.36	0.30
	Avg[1]	0.93	0.94	0.98	0.36	0.32	0.26
poly3	Fit[1]	1.10	1.14	1.21	0.42	0.37	0.31
	Avg[1]	0.97	0.98	1.02	0.37	0.32	0.27
poly4	Fit[1,2]	1.11	1.15	1.23	0.43	0.38	0.32
	Avg[2]	0.98	1.00	1.04	0.38	0.33	0.28
Trapezoid	Fit[1]	1.16	1.19	1.26	0.44	0.38	0.32
	Avg[1]	1.11	1.13	1.14	0.42	0.36	0.30
Triangle	Fit[1]	1.12	1.14	1.22	0.42	0.37	0.31
	Avg[1]	1.10	1.14	1.21	0.42	0.37	0.31
Bass-II[1,2]	–	1.08	0.94	0.97	0.38	0.34	0.26

Note: Double boxes denote the best values in each column, while single boxes denote the second-best values. Superscripts *1* and *2* denote that the best results for the curve-fitting methods were obtained by fitting the parameters to the entire *GenerateAverage*/*GenerateFit* curves per cluster and only to the part after the introduction weeks, respectively.

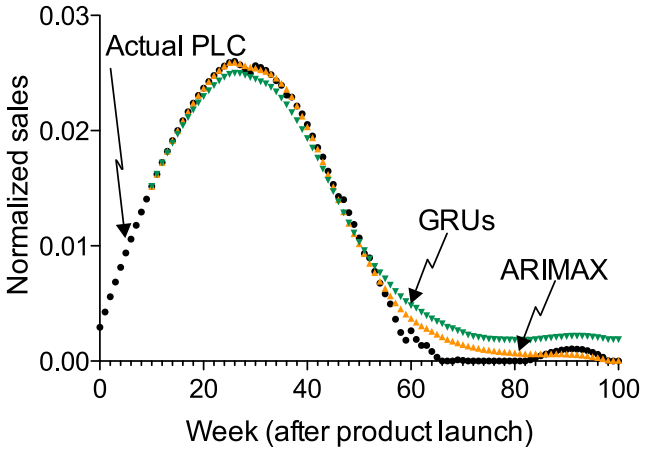


Fig. 10. Comparison of forecasting results of ARIMAX and GRUs (SKU 112 from the Dell data set, with 10 weeks of introduction).

GRUs. Furthermore, ARIMAX shows a clear end of PLC as the sales drop to zero, while GRUs shows that the life cycle is longer than expected. In Fig. 11, we illustrate the forecasting results of ARIMAX and Bass-II on another example SKU with 10 weeks of post-launch sales data. As can be seen, the results show that the forecasting performance of ARIMAX is superior to Bass-II, which produces a PLC with a lower peak and extended selling period, as opposed to the actual PLC and the results of ARIMAX.

Although Hu et al. (2018) did not report actual forecast errors, which would have allowed us to compare our results with theirs, it is evident that the relative performance of the curve families differs slightly. In their study of pre-launch forecasts, even though poly2 performed well, Trapezoid and Triangle performed best. Unlike Hu et al. (2018), we find that the Bass model gives very good results. This difference in relative performance may well be explained by differences in the smoothing step or the

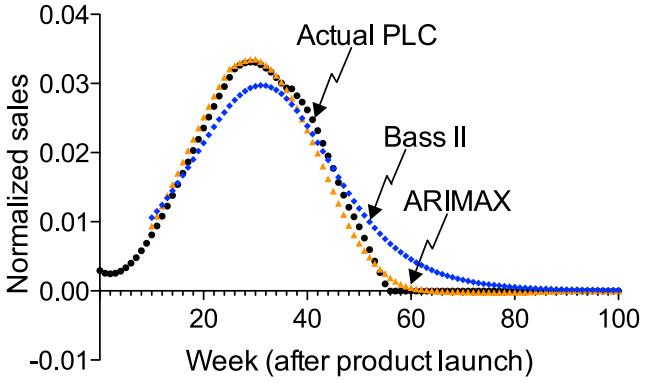


Fig. 11. Comparison of forecasting results of ARIMAX and Bass-II (SKU 24 from the Dell data set, with 10 weeks of introduction).

fact that we consider the Dell products' entire life-cycle length, rather than truncate the end of the life cycles like Hu et al. (2018). Also, it is important to note that we do not normalize the selling period and therefore our error calculations penalize the results if the methods do not predict the life-cycle length accurately.

It is interesting to see, however, that while all considered methods exhibit improved performance in terms of SAE when more weeks of sales are included, the quantitative forecasting methods do not always benefit from time-series clustering. The forecasts generated by ARIMAX show substantially worse performance when using clustering, whereas the DNNs with clustering often have lower forecast errors (and, hence, higher accuracy), with reductions in MASE (SAE) of up to 7.63% (18.6%), 4.31% (9.38%), and 3.65% (3.92%) under LSTM, GRUs, and CNNs, respectively. The intuition behind this is that after clustering, the variance between the sales of products in the clusters is minimized, removing possible noise coming from PLCs that are different and that could affect the forecasting accuracy. The reductions are rather modest, as the Dell data set is composed of only 170 SKUs, so separating the data into clusters reduces the amount of data available for the DNN algorithms to train. We should therefore expect greater improvements in accuracy by clustering when applying our framework to the Retailer X data set, as it is almost five times larger.

6.2. Retailer X data set

We now repeat the clustering and forecasting steps for the second data set – Retailer X – to analyze the performance of our proposed framework on a second, much larger data set which consists of almost 850 SKUs and features irregular PLCs such as those with two maturity phases. The accuracy of post-launch forecasts is evaluated at six, eight, and 10 weeks after a new product is launched. We chose 10 weeks as the maximum, since the majority of products have already started their decline stage by then.

The results in terms of forecast accuracy are displayed in Table 4 and show that the MASE generally does not follow any trend when more weeks are introduced, whereas the SAE virtually always decreases as more data become available. While the latter confirms that the considered forecasting methods typically perform better with more data, the improvement is not reflected in the MASE, due to the fact that we introduce only two more weeks at a time. It is now more apparent that clustering improves the forecast accuracy of the quantitative methods, with average MASE (SAE) improvements of 14.67% (19.16%), 8.97% (20.18%), 5.19% (12.54%), and 20.89% (21.31%) under LSTM, GRUs, CNNs, and ARIMAX, respectively, which is in line with the best results obtained for the Dell data set for the DNN methods. Due to the increased amount of data for Retailer X, the reduction in forecasting errors over no clustering is much more substantial than for the Dell data set, so clustering can be regarded as an important step, especially when historical PLC data are plentiful.

The forecast errors in Table 4 support the results of the Dell data set, with ARIMAX still generating the best results, but the difference between ARIMAX and the three DNNs is now reduced. In fact, when compared with the best-performing DNN – again, either LSTM or GRUs – the percentage reductions in MASE (SAE) achieved by ARIMAX with clustering are 9.24% (6.98%), 11.02% (10.26%), and 10.29% (9.09%) for six, eight, and 10 input weeks, respectively. On the other hand, when compared with CNNs, ARIMAX again achieved higher improvements of 26.53% (24.53%), 23.91% (20.45%), and 28.65% (23.07%), respectively. The standard deviation within the time-series clusters of Retailer X is higher than that of Dell's, meaning the products have less similar PLC curves and the linear dependencies between the PLCs are minimized, which may explain why the forecast errors for ARIMAX are almost always higher now.

It can also be observed from Table 4 that although the PLC shape-based methods still perform relatively well, they are generally outperformed by the DNN methods in terms of SAE. Comparing the performance of ARIMAX to the best-performing PLC shape-based method – either Bass-I (Avg) or Bass-II – the percentage reductions in MASE (SAE) achieved by ARIMAX are 1.82% (24.53%) 10.26% (28.57%), and 1.61% (33.33%) for six, eight, and 10 input weeks, respectively. The improved comparative performance of the ML methods may again be explained by the PLC curves in the Retailer X data set, which are less similar and thus show a higher degree of variability than

Table 4
Forecast errors for new products for different weeks after introduction (Retailer X data).

Quantitative method	Time-series clustering	MASE			SAE		
		6 weeks	8 weeks	10 weeks	6 weeks	8 weeks	10 weeks
LSTM	No	1.40	1.46	1.52	0.50	0.46	0.46
	Yes	1.19	1.19	1.36	0.43	0.39	0.33
GRUs	No	1.41	1.46	1.50	0.52	0.49	0.45
	Yes	1.47	1.18	1.32	0.45	0.39	0.33
CNNs	No	1.65	1.61	1.56	0.57	0.52	0.46
	Yes	1.47	1.38	1.71	0.53	0.44	0.39
ARIMAX	No	1.26	1.53	1.47	0.47	0.44	0.42
	Yes	1.08	1.05	1.22	0.40	0.35	0.30
Base curve family	Cluster curve generation	MASE			SAE		
		6 weeks	8 weeks	10 weeks	6 weeks	8 weeks	10 weeks
Bass-I	Fit[2]	1.20	1.20	1.30	0.56	0.50	0.49
	Avg[1,2]	1.10	1.17	1.24	0.53	0.49	0.45
poly2	Fit[1]	1.40	1.46	1.57	0.67	0.62	0.58
	Avg[2]	1.15	1.40	1.54	0.54	0.50	0.47
poly3	Fit[2]	2.11	2.85	3.30	0.99	0.83	0.71
	Avg[2]	1.14	1.39	1.54	0.54	0.50	0.46
poly4	Fit[2]	1.61	1.82	2.14	0.78	0.71	0.64
	Avg[1,2]	1.13	1.38	1.53	0.54	0.51	0.46
Trapezoid	Fit[2]	1.29	1.36	1.34	0.61	0.58	0.51
	Avg[1,2]	1.29	1.30	1.25	0.58	0.54	0.47
Triangle	Fit[2]	1.28	1.36	1.39	0.60	0.55	0.50
	Avg[2]	1.20	1.25	1.26	0.56	0.51	0.47
Bass-II[2]	–	1.11	1.17	1.24	0.55	0.49	0.47

Note: Double boxes denote the best values in each column, while single boxes denote the second-best values. Superscripts *1* and *2* denote that the best results for the curve-fitting methods were obtained by fitting the parameters to the entire *GenerateAverage*/*GenerateFit* curves per cluster and only to the part after the introduction weeks, respectively.

those of Dell, and some of which even follow irregular life cycles with two maturity stages.

It is interesting to note at this point that the performance gap between ARIMAX and the DNNs reduces as both the variability in the historical PLC data increases and the similarity between past products' life cycles decreases. Since PLC curves of new products are not smooth but rather tend to fluctuate substantially, we smoothed the historical PLC curves before generating forecasts (see Fig. 4). In the next sub-section, we introduce noise to the (smoothed) input data for the new product to portray the difference in the accuracy of the quantitative forecasting techniques – DNNs with clustering and ARIMAX without clustering – for situations with various degrees of sales fluctuations. We do not test the robustness of the shape-based methods because, apart from Bass-II, adding noise to the sales of the newly launched product does not change the fact that we fit the base curve shapes only to the PLC curves of past products (albeit truncated at the beginning under the superscript *2* strategy), which are unaffected by this noise.

7. Robustness of the quantitative methods

7.1. Impact of adding Gaussian white noise

Until now we have used smoothed data from the first weeks of sales as input to the different forecasting methods for the new products. The first few weeks, however, may not be sufficient to generate an optimal fitted curve that would represent the entire new PLC. So unsmoothed or slightly smoothed sales of the new product might be used as input into the forecasting algorithms.

To test how robust the quantitative methods are to noise in the input data for new products, we introduce noise with different standard deviations and evaluate the impact on the forecast accuracy of each method. In particular, Gaussian noise with zero mean ($\mu = 0$) and different levels of standard deviation (σ) is introduced to disrupt the sales during the first few weeks after the new product is launched. Introducing Gaussian white noise (GWN) is a common strategy to test the robustness of different forecasting techniques, and we use it as our basic noise model. It is generally known that DNNs are prone to overfitting, so adding noise will provide further validation of this characteristic. We add noise after the smoothing step to be able to measure and control its level, because introducing noise to the unsmoothed sales data can have the opposite effect by decreasing the actual amount of noise in the input.

Let $\tilde{D}^{a,i} = (\tilde{D}_t^{a,i})_{t=1}^{\tau}$ denote the vector of the smoothed sales already encountered for product i. The length of the vector corresponds to the number of weeks we choose to introduce, with each component $\tilde{D}_t^{a,i}$ representing the sales for week t, where $t \in \{1, 2, \ldots, \tau\}$ and τ is the number of weeks considered after a new product's launch. Then, the vector of disrupted weekly input sales of the new product i after adding GWN, $\tilde{y}^{(i)}$, is given by

$$\tilde{y}^{(i)} = \tilde{D}^{a,i} + \epsilon, \tag{6}$$

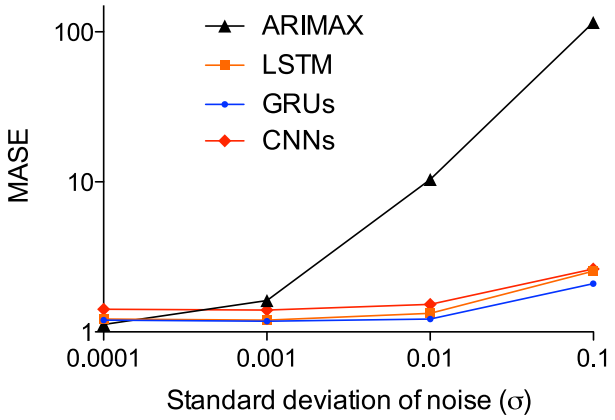


Fig. 12. Forecast errors (MASEs) after adding Gaussian white noise to Dell data (with a log-scale vertical axis).

where ϵ is a vector of the same length as $\tilde{D}^{a,i}$ that consists of components that are chosen randomly (i.i.d.) from a zero-mean normal distribution: $\epsilon_t \sim \mathcal{N}(0, \sigma^2)\ \forall t \in \{1, 2, \ldots, \tau\}$.

The results of the robustness testing are reported in Fig. 12. We chose to test standard deviations $\sigma \in \{0.0001, 0.001, 0.01, 0.1\}$ and considered $\tau = 20$ weeks of product introduction. The results show that all the methods are robust to GWN for σ-values up to 0.001. However, when σ is increased to 0.01, the average MASE reported for ARIMAX is substantially increased. The three DNNs, by contrast, are resistant and therefore robust to this noise level. Although introducing noise with a σ of 0.1 causes the forecast errors for all methods to increase, only ARIMAX's performance deteriorates drastically and its forecast error increases exponentially. The intuition behind this behavior is that ARIMAX detects linear relationships between the sales data, whereas the DNN methods detect nonlinear relationships. Because noise standard deviation is inversely correlated to linearity, ARIMAX's performance decreases substantially.

The robustness results indicate that ARIMAX is considerably more sensitive to high levels of noise in the sales data than the three DNNs under consideration. When compared with the worst-performing DNN – CNNs – the forecast errors (MASEs) achieved by ARIMAX are 15.71%, 578.43%, and 4291.63% higher for standard deviations 0.001, 0.01, and 0.1, respectively. On the other hand, when compared with either LSTM or GRUs, ARIMAX achieved even higher increases of 36.14%, 715.64%, and 4923.62%, respectively. Different levels of noise can represent different sales patterns that may occur in real-world situations. In fact, we also tested the forecasting methods using unsmoothed sales as input for the new products and obtained results very similar to those reported above for low levels of σ. This similarity suggests that low noise levels correspond to sales fluctuations that can be reasonably expected to be present in real-world sales data. Higher values of σ can represent larger unexpected fluctuations, such as sharp spikes and/or drops in sales, that may occur due to economic booms and busts, recessions, wars, and pandemics.

7.2. Effect of incorrect cluster assignment

Although we have used integration and DTW to assist managers in placing the new product in a cluster, the results in Fig. 6 show that, on average, both cluster assignment methods applied provide only around 60% correct assignments given the input weeks considered in our study. Until now we have assumed that we know which cluster each new product belongs to when forecasting. In this section, we assign the new products to (in)correct clusters and report the forecasting errors in terms of the MASE. In the tables presented in this section, the rows represent the clusters the products being forecasted belong to, and the columns represent the clusters the new products have been (intentionally incorrectly) assigned to. All results are obtained using 10 weeks of input data, which corresponds to about 60% correct cluster assignment for both integration and DTW (see Fig. 6). The forecast errors are calculated for 15% of the data of each cluster and averaged over three repetitions, using the same set of randomly selected products for the different methods in each repetition to ensure the comparability of the results. The MASE results obtained using all products in the Dell data set for forecasting, rather than solely those of the assigned cluster, are reported for comparison in online Appendix K.

The results in Table 5 show an increase in the average MASE for ARIMAX over all clusters with incorrect cluster assignment when compared to the highest possible accuracy achieved by either forecasting with correct cluster assignment (diagonal entries in Table 5) or forecasting without clustering (see the ARIMAX column of Table K.13).

Note that although the results indicate that incorrectly assigning the products of cluster 1 (to clusters 2, 3, or 4) results in equally accurate forecasts as correct assignments (to cluster 1), reporting results to more than two decimal places would actually show that using all historical data (i.e. without clustering) produces slightly more accurate forecasts. It can also be seen that incorrectly assigning a product to cluster 1 results in substantial increases in MASE values. This is because the properties of cluster 1 are very different, due to the high initial sales

Table 5
Average MASE results for (in)correct cluster assignment using ARIMAX on Dell data.

From	To			
	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	1.77	1.77	1.77	1.77
Cluster 2	91.56	0.86	0.87	0.86
Cluster 3	23.15	1.86	1.07	1.07
Cluster 4	38.68	1.75	1.05	0.85

Table 6
Average MASE results for (in)correct cluster assignment using LSTM on Dell data.

From	To			
	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	1.84	1.78	1.50	1.72
Cluster 2	1.17	0.97	1.45	1.16
Cluster 3	1.91	1.72	1.20	1.47
Cluster 4	1.58	1.19	1.39	0.92

Table 7
Average MASE results for (in)correct cluster assignment using GRUs on Dell data.

From	To			
	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	1.87	1.77	1.50	1.73
Cluster 2	1.17	0.98	1.45	1.16
Cluster 3	2.02	1.80	1.20	1.48
Cluster 4	1.61	1.20	1.39	0.93

Table 8
Average MASE results for (in)correct cluster assignment using CNNs on Dell data.

From	To			
	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	1.78	1.86	1.46	1.63
Cluster 2	1.23	1.01	1.46	1.28
Cluster 3	2.29	1.81	1.42	1.45
Cluster 4	1.73	1.23	1.42	1.02

volumes of the PLCs and short life cycles, and since ARIMAX captures linear relationships, the errors considerably increase if the products in the cluster are very different from the new products' initial sales. Assuming forecasters do not misplace a product in cluster 1 given the distinct characteristics of the clusters, ARIMAX with an incorrect cluster assignment results in an average MASE of 1.42, which is substantially lower than in the three worst-case incorrect cluster assignments.

Tables 6, 7 nd 8 show the results of (in)correct cluster assignment for LSTM, GRUs, and CNNs, respectively. It can be observed that, except for cluster 1, an incorrect cluster assignment virtually always results in higher MASE values compared to a correct assignment. A correct cluster assignment, however, does not necessarily lead to the most accurate forecasts. In fact, unlike for products from clusters 2, 3, and 4, forecasts for products from cluster 1 are generally more accurate when all PLCs in the data set are used for forecasting (see Table K.13). Somewhat counter-intuitively, correctly assigning products from cluster 1 (to cluster 1) generally results in poorer forecasts than incorrectly assigning them to clusters 2–4, which may be explained by the small size and the distinct characteristics of cluster 1. Although the forecast accuracy of DNNs also typically decreases through incorrect assignments, the percentage increases in MASE values are lower than for ARIMAX, with CNNs showing the least percentage increase, which suggests that their performance is more robust to incorrect cluster assignments.

It is interesting to note that the considered DNNs share the same worst-case incorrect cluster assignment across clusters. In fact, as can be seen from Tables 6–8, for all three DNNs, the worst incorrect assignment in terms of average MASE is found by assigning products from clusters 1, 2, 3, and 4 to clusters 2, 3, 1, and 1, respectively. Based on the results of this robustness analysis, it can be argued that if forecasters are not sure to which cluster a new product belongs, they may want to consider using all the old PLCs rather than risking an incorrect cluster assignment that might lead to poor forecasts. Indeed, as shown in this section, forecasting without clustering often results (especially for products from cluster 1) in lower forecast errors than those obtained using incorrect cluster assignments. Forecasters can then apply the ML-based forecasting framework proposed in this paper and follow all the steps, apart from the clustering step, to obtain more accurate sales forecasts of newly launched, short-lived products.

8. Conclusions

In this paper, we developed a framework that uses quantitative methods to forecast sales of new, but not completely new, products with short life cycles after their launches. To accomplish this, our framework involves a number of important steps based on state-of-the-art techniques. Firstly, we apply smoothing to obtain representative product life cycle (PLC) sales and then use time-series clustering to group similar products. In order to assign the newly launched product to one of the clusters, we consider two alternative quantitative methods – integration and dynamic time warping (DTW). Subsequently, we perform data augmentation, a common approach in machine learning (ML), to generate artificial data points to support the training of the forecasting algorithms. In addition to three deep neural networks (DNNs) – long short-term memory (LSTM), gated recurrent units (GRUs), and convolutional neural networks (CNNs) – we apply a traditional statistical approach, autoregressive integrated moving average with exogenous variables (ARIMAX), in our framework's forecasting step. We also apply three forecasting methods that are based on fitting families of curves – Bass diffusion, polynomial, and piece-wise linear curves — to historical PLC shapes.

We illustrated the applicability of our framework using the publicly available Dell data set comprising complete PLC order history for 170 products, and we additionally evaluated the comparative performance of the forecasting methods using a second, much larger data set – from Retailer X – that includes customer orders for 843 complete PLCs. We investigated the accuracy of both integration and DTW in correctly assigning a new product to its cluster, and found that the effectiveness of the latter was somewhat limited. In our empirical analysis, we found that ARIMAX gave the best forecasting results for the three different numbers of input weeks considered and, for our larger data set, showed that clustering generally resulted in substantially lower forecast errors. In fact, we showed that, when compared with the best-performing DNN – either LSTM or GRUs – the forecast errors (MASEs) achieved by ARIMAX using Dell data were 24.32%, 23.64%, and 21.20% lower for 10, 15, and 20 input weeks, respectively. On the other hand, when compared with the best-performing PLC shape-based method – either poly2 (Avg) or Bass-II – ARIMAX achieved comparatively lower reductions of 9.68%, 10.64%, and 11.34%, respectively. We obtained consistent results for our larger data set, but found that the performance gap between the DNNs and ARIMAX was reduced and that the former generally outperformed the PLC shape-based methods.

We also investigated the robustness of the quantitative forecasting methods to noise in the input data and to an incorrect cluster assignment. We added Gaussian white noise with different levels of standard deviation to represent different sales patterns that may occur in real-life situations and found that ARIMAX's performance deteriorated drastically as the level of noise increased, whereas the three DNN methods' forecast accuracy remained relatively unaffected. In fact, we found that, when compared with the worst-performing ML method – CNNs – the forecast errors (MASEs) achieved by ARIMAX were more than 15%, 578%, and 4291% higher for standard deviations of 0.001, 0.01, and 0.1, respectively. This means that the three DNNs are considerably more robust to noise in data sets and, as such, more suitable for forecasting sales of newly launched, short-lived products when there are sufficiently large demand fluctuations such as sudden spikes and drops in sales. We assigned new products to incorrect clusters to analyze the resulting impact on forecasting results and found that the errors generally increased considerably. Our results suggest that forecasting may be carried out without the clustering step using all data if it is unclear to which cluster a new product belongs.

The framework proposed in this paper enables practitioners to use quantitative methods to forecast demand in data-scarce situations, and the managerial insights provided support practitioners to decide when to apply ML-based forecasting methods and when to apply traditional methods. In fact, we demonstrated how state-of-the-art techniques that are simple and readily applicable can be combined in a powerful framework in order to enable companies to exploit the untapped potential of quantitative forecasting methods when there is limited availability of historical sales data. For example, integration for cluster assignment – by determining the area under the curve – and data augmentation for the generation of sufficient training data – by performing interpolation – are both simple and straightforward methods that can be readily applied by practitioners.

There are several important directions for future research. For example, our framework could be used in tandem with existing prediction tools that provide pre-launch forecasts to improve overall accuracy by forecast updating. Alternatively, by assuming that the cluster to which a new product belongs is known before the first demand is realized, our framework could also be used to generate a forecast before a product's launch date. It could also be interesting to evaluate other quantitative methods, such as exponential smoothing or multi-layer perceptron, to incorporate additional information such as pricing and product reviews and to explore the application of our proposed framework to forecast sales of old or long-lived products.

Acknowledgments

The authors wish to thank the two anonymous reviewers, an anonymous associate editor, and Prof. Fotios Petropoulos (editor) for their many helpful comments and valuable suggestions. The authors are also grateful to Prof. Jason Acimovic for sharing the Dell data set and to members of Retailer X (disguised company name) for sharing the Retailer X data set.

Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ijforecast.2022.09.005.

References

Acimovic, J., Erize, F., Hu, K., Thomas, D. J., & Van Mieghem, J. A. (2018). Product life cycle data set: Raw and cleaned data of weekly orders for personal computers. *Manufacturing & Service Operations Management*, *21*(1), 171–176.

Baardman, L., Levin, I., Perakis, G., & Singhvi, D. (2018). Leveraging comparables for new product sales forecasting. *Production and Operations Management*, *27*(12), 2340–2343.

Basallo-Triana, M. J., Rodríguez-Sarasty, J. A., & Benitez-Restrepo, H. D. (2017). Analogue-based demand forecasting of short life-cycle products: A regression approach and a comprehensive assessment. *International Journal of Production Research*, *55*(8), 2336–2350.

Bass, F. M. (1969). A new product growth for model consumer durables. *Management Science*, *15*(5), 215–227.

Box, G. E. P., & Tiao, G. C. (1975). Intervention analysis with applications to economic and environmental problems. *Journal of the American Statistical Association*, *70*(349), 70–79.

Burruss, J., & Kuettner, D. (2002). Forecasting for short-lived products: Hewlett-Packard's journey. *The Journal of Business Forecasting Methods & System*, *21*(4), 9–14.

Cao, Q., Ewing, B. T., & Thompson, M. A. (2012). Forecasting wind speed with recurrent neural networks. *European Journal of Operational Research*, *221*(1), 148–154.

Cecere, L. (2013). New products: More costly and more important. *Forbes*, (Accessed 16 May 2020).

Cooper, R. G., & Edgett, S. J. (2012). Best practices in the idea-to-launch process and its governance. *Research-Technology Management*, *55*(2), 43–54.

Dean, J. (1976). Pricing policies for new products. *Harvard Business Review*, (Accessed 21 February 2019).

DeVries, T., & Taylor, G. W. (2017). Dataset augmentation in feature space. In *International conference on learning representations, ICLR 2017 - workshop track*.

Fallah Tehrani, A., & Ahrens, D. (2016). Enhanced predictive models for purchasing in the fashion field by using kernel machine regression equipped with ordinal logistic regression. *Journal of Retailing and Consumer Services*, *32*, 131–138.

Fildes, R., & Goodwin, P. (2007). Against your better judgment? How organizations can improve their use of management judgment in forecasting. *Interfaces*, *37*(6), 570–576.

Golder, P. N., & Tellis, G. J. (2004). Growing, growing, gone: Cascades, diffusion, and turning points in the product life cycle. *Marketing Science*, *23*(2), 207–218.

Hamilton, L., & Webster, P. (2012). *The international business environment* (2nd ed). Oxford University Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hu, K., Acimovic, J., Erize, F., Thomas, D. J., & Van Mieghem, J. A. (2018). Forecasting new product life cycle curves: Practical approach and empirical analysis. *Manufacturing & Service Operations Management*, *21*(1), 66–85.

Hu, L., Wang, C., Ye, Z., & Wang, S. (2021). Estimating gaseous pollutants from bus emissions: A hybrid model based on GRU and XGBoost. *Science of the Total Environment*, *783*, Article 146870.

Iwana, B. K., & Uchida, S. (2021). An empirical survey of data augmentation for time series classification with neural networks. *PLoS ONE*, *16*(7), e0254841.

Javed, A., Lee, B. S., & Rizzo, D. M. (2020). A benchmark study on time series clustering. *Machine Learning with Applications*, *1*, Article 100001.

Kahn, K. B. (2002). An exploratory investigation of new product forecasting practices. *Journal of Product Innovation Management*, *19*(2), 133–143.

Kahn, K. B. (2014). Solving the problems of new product forecasting. *Business Horizons*, *57*(5), 607–615.

Kraus, M., Feuerriegel, S., & Oztekin, A. (2020). Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research*, *281*(3), 628–641.

Kumar, S., Hussain, L., Banarjee, S., & Reza, M. (2018). Energy load forecasting using deep learning approach-LSTM and GRU in spark cluster. In *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)* (pp. 1–4).

Kurawarwala, A. A., & Matsuo, H. (1996). Forecasting and inventory management of short life-cycle products. *Operations Research*, *44*(1), 131–150.

Lee, C. .-Y., & Lee, M. .-K. (2017). Demand forecasting in the early stage of the technology's life cycle using a Bayesian update. *Sustainability*, *9*(8), 1378.

Liu, C., Hou, W., & Liu, D. (2017). Foreign exchange rates forecasting with convolutional neural network. *Neural Processing Letters*, *46*, 1095–1119.

Lu, C. .-J., & Kao, L. .-J. (2016). A clustering-based sales forecasting scheme by using extreme learning machine and ensembling linkage methods with applications to computer server. *Engineering Applications of Artificial Intelligence*, *55*, 231–238.

Mahajan, V., Muller, E., & Bass, F. M. (1990). New product diffusion models in marketing: A review and directions for research. *Journal of Marketing*, *54*(1), 1–26.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, *36*(1), 54–74.

Meade, N., & Islam, T. (2006). Modelling and forecasting the diffusion of innovation – A 25-year review. *International Journal of Forecasting*, *22*(3), 519–545.

Mišić, V. V., & Perakis, G. (2019). Data analytics in operations management: A review. *Manufacturing & Service Operations Management*, *22*(1), 158–169.

Özkoç, E. E. (2020). Clustering of time-series data. In *(Ed.), Data mining - methods, applications and systems*. IntechOpen.

Paliari, I., Karanikola, A., & Kotsiantis, S. (2021). A comparison of the optimized LSTM, XGBOOST and ARIMA in time series forecasting. In *12th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1–7).

Park, D. S., Chan, W., Zhang, Y., Chiu, C. .-C., Zoph, B., Cubuk, E. D., et al. (2019). SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019* (pp. 2613–2617).

Peres, R., Muller, E., & Mahajan, V. (2010). Innovation diffusion and new product growth models: A critical review and research directions. *International Journal of Research in Marketing*, *27*(2), 91–106.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., et al. (2022). Forecasting: Theory and practice. *International Journal of Forecasting*, *38*(3), 705–871.

Qi-zhi, S. (2007). Forecasting for products with short life cycle based on improved BASS model. *Industrial Engineering and Management*.

Rink, D. R., & Swan, J. E. (1979). Product life cycle research: A literature review. *Journal of Bussiness Research*, *7*(3), 219–242.

Rogers, E. M. (1962). *Diffusion of innovations*. New York: Free Press of Glencoe.

Rudin, C., & Carlson, D. (2019). The secrets of machine learning: Ten things you wish you had known earlier to be more effective at data analysis. *INFORMS Tutorials in Operations Research*.

Sanders, N. R., & Manrodt, K. B. (2003). The efficacy of using judgmental versus quantitative forecasting methods in practice. *Omega*, *31*(6), 511–522.

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, *Part B*, Article 105524.

Stellwagen, E., & Tashman, L. (2013). ARIMA: The models of Box and Jenkins. *Foresight: The International Journal of Applied Forecasting, International Institute of Forecasters*, (30), 28–33.

Szozda, N. (2010). Analogous forecasting of products with a short life cycle. *Decision Making in Manufacturing and Services*, *4*(2), 71–85.

Thomassey, S., & Happiette, M. (2007). A neural clustering and classification system for sales forecasting of new apparel items. *Applied Soft Computing*, *7*(4), 1177–1187.

van Steenbergen, R. M., & Mes, M. R. K. (2020). Forecasting demand profiles of new products. *Decision Support Systems*, *139*, Article 113401.

Vernon, R. (1966). International investment and international trade in the product cycle. *The Quaterly Journal of Economics*, *80*(2), 190–207.

Yildiz, B., Bilbao, J. I., & Sproul, A. B. (2017). A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renewable and Sustainable Energy Reviews*, *73*, 1104–1122.

Zhang, X., Shen, F., Zhao, J., & Yang, G. (2017). Time series forecasting using GRU neural network with multi-lag after decomposition. In D. Liu, S. Xie, Y. Li, D. Zhao, & E. El-Alfy (Eds.), *vol. 10638, Neural Information Processing. ICONIP 2017. Lecture Notes in Computer Science*. Springer, Cham.

Zhu, K., & Thonemann, U. W. (2004). An adaptive forecasting algorithm and inventory policy for products with short life cycles. *Naval Research Logistics*, *51*(5), 633–653.