Retail sales forecasting with meta-learning

Shaohui Ma[a,1]

Robert Fildes[b]

[a] School of Business, Nanjing Audit University, China, 211815

[b] Centre for Marketing Analytics and Forecasting, Lancaster University, UK, LA1 4YX

January 2020 revised March 2020

[1] Corresponding author at: School of Business, Nanjing Audit University, Nanjing, 211815, China. E-mail address: shaohui.ma@nau.edu.cn (Shaohui Ma).

Abstract

Retail sales forecasting often requires forecasts for thousands of products for many stores. We present a meta-learning framework based on newly developed deep convolutional neural networks, which can first learn a feature representation from raw sales time series data automatically, and then link the learnt features with a set of weights which are used to combine a pool of base-forecasting methods. The experiments which are based on IRI weekly data show that the proposed meta learner provides superior forecasting performance compared with a number of state-of-art benchmarks, though the accuracy gains over some more sophisticated meta ensemble benchmarks are modest and the learnt features lack interpretability. When designing a meta-learner in forecasting retail sales, we recommend building a pool of base-forecasters including both individual and pooled forecasting methods, and target finding the best combination forecasts instead of the best individual method.

Keywords: Forecasting; big data; retail sales forecasting; machine learning; forecasting many time series; meta learning; deep learning

1. Introduction

Retail sales forecasting is often concerned with generating forecasts for a large number of products across many stores over a short forecasting horizon. Sales forecasts are the essential inputs to many managerial decisions, such as pricing, store space allocation, listing/delisting, ordering and inventory management for an item. Forecasts also provide the basis for distribution and replenishment plans. The ability of retail managers to estimate the expected sales quantity at the SKU (Stock Keeping Unit) × store level over the short term should lead to improved customer satisfaction, reduced waste, increased sales revenue and more effective and efficient distribution (Fildes, Ma, & Kolassa, 2020). A good sales forecasting system also allows retailers to simulate the results of their different promotional mixes, and then optimize the promotional schedules (Levy, Grewal, Kopalle, & Hess, 2004).

Large retail chains accumulate huge amount of sales data through their POS (Point Of Sale) machines, however at store x item level the data is often scarce, as the assortments in each store can change rapidly due to the increasingly competitive retail market environment. Retail product demands are also driven by many factors, e.g., price changes, promotions, special events, seasons, holidays, and even weather. As a result, store item level sales data are characterized by high volatility and skewness, multiple seasonal cycles especially when combined with 'special days' (e.g., bank holidays), their often large volume, alternatively intermittence with zero sales frequently observed at store level, together with high dimensionality in any explanatory variable space. These issues make accurately forecasting item level sales a difficult task.

Many methods have been proposed to improve the sales forecasting accuracy for retail products. But most studies have aimed at proposing a universal forecasting method which is used for all the sales time series under their study (Fildes, et al.,, 2020). None of research in retail forecasting has tried to use various forecasting methods according to the data characteristics of different store items. In this research, we propose a meta-learning framework based on newly developed deep convolutional neural networks, which first learns from knowledge of the forecasting performance of a given combination of base-forecasting methods (forecasters) as this relates to the characteristics of the data used to fit these base-forecasters, and then uses that knowledge to generate an optimal ensemble (combination) of forecasts to forecast the sales of a product according to its specific data history. Meta-learning methods therefore allows different ensemble models to be used to forecast different products at different periods of time, in contrast to relying on one model to forecast sales of all products in all time periods.

The contributions of this paper are four fold. First, this is the first to empirically evaluate the performance of meta-learning in a retail product sales forecasting setting. Second, we propose a novel meta-learner which can learn a feature representation from raw time series data automatically. Third, we explore the impacts of the constituents of its base-forecasters on the forecasting performance of the meta-learner. Fourth, we investigate the value of extracting features from external potential influences,

in addition to the sales time series data on the forecasting performance of the proposed meta-learner. Overall, what we achieve here is a composite meta-learner providing improved accuracy performance, which can be applied automatically in complex problem areas such as that faced in retailing where the scale of the problem makes automatic method selection a key requirement.

The outline of the paper is as follows. In Section two, we review related studies and introduce our innovations. In Section three we discuss associated methodological issues in our proposed solution. In Section four, we describe the data, introduce the experimental design and forecasting accuracy measures. We then present the empirical results in Section five. In the last section, we discuss the findings, offering conclusions as to forecasting practice and further academic research.

2. Related research

2.1. Retail sales forecasting

Much effort has been devoted over the past several decades to the development and improvement of sales forecasting models in retail (see Fildes et al. (2020) for a deeper survey). The basic product sales forecasting methods are based on univariate forecasting models using only the past sales history. The techniques used in retail range from the traditional time series techniques, such as simpler moving averages, the exponential smoothing family or the more complicated Box–Jenkins ARIMA approach (Kalaoglu et al., 2015), Fourier analysis (Fumi, Pepe, Scarabotti, & Schiraldi, 2013), to state space models (Ramos, Santos, & Rebelo, 2015).

Another stream of studies uses a model-based forecasting system to forecast product sales by directly taking into account promotional (and other) information. These methods are usually based on multiple linear regression models or more complex econometric models whose exogenous inputs correspond to seasonality, calendar events, weather conditions, price, and promotion features, as surveyed in Fildes et al. (2020). While some of these promotional inputs can be considered endogenous, for forecasting purposes nothing is typically gained from adopting a systems approach (Allen & Fildes, 2001). Overall, the multivariate studies show substantial accuracy improvements for SKU level forecasts over univariate benchmarks.

Nonlinear models include traditional nonlinear regressions, non- or semi-parametric regressions, and soft computing techniques. The models used include Back Propagation Neural Networks (Aburto & Weber, 2007; Ainscough & Aronson, 1999), Regression Trees (Gür Ali, Sayin, van Woensel, & Fransoo, 2009), Support Vector Machines (Gür Ali & Yaman, 2013; Pillo, Latorre, Lucidi, & Procacci, 2016), Bayesian P-splines (Lang, Steiner, Weber, & Wechselberger, 2015), and recurrent neural networks (Salinas, Flunkert, Gasthaus, & Januschowski, 2019), etc. Most published research has found improvements in forecasting accuracy by using nonlinear models over linear regressions. But the positive evidence is probably enhanced by publication bias, which may be amplified by apparent poor

forecast evaluation practice in the machine learning community, along with a certain amount of hype. So wide ranging evidence of the benefits of machine learning algorithms is needed if we are to accept the hype that both researchers and software companies have generated.

So far, existing research in retail sales forecasting has focused on using a universal forecasting method that can be applied to all the products under study. However, according to the no-free-lunch theorem of Wolpert and Macready (1997), there is no guarantee that any method, however complex it may be, performs better for a different set of series than another method; this implies that it is unlikely that one single method will dominate others for all products and all future time periods. Evidence on relative performance is specific to the application with retail sales (as here) having unique characteristics that do not correspond to the massive 'competition' studies (Makridakis & Petropoulos, 2020). This underlines the importance of method selection to match the problem characteristics. Within the broad forecasting community there has been relatively little research that has explored the benefits of different approaches to selection, comparing individual selection and combination versus aggregate selection and combination but see (Fildes & Petropoulos, 2015). The F&P study found accuracy benefits from selecting a range of methods to match the data and series forecasting performance characteristics, which suggests there may be benefits to be had from extending this approach to retailing and including a wider range of methods and series features that capture the characteristics of retail data. In this paper, we contribute to the retail sales forecasting literature by proposing a meta-learning framework which can forecast each sales series with a different forecasting method designed according to the characteristics of the SKU's sales series and its particular influential factors.

2.2 Forecasting many time series with meta-learning

The term meta-learning was first adopted in the context of forecasting many time series by Prudêncio and Ludermir (2004). A meta-learning framework for forecasting many time series usually consists of three components: a set of features extracted from the time series, a pool of base-forecasters, and a meta-learner. The meta-learner is used to learn the meta-knowledge that may be captured by linking the features summarizing characteristics of the time series to the forecasting performance of the base forecasters, and then the learnt knowledge is used to select an optimal forecasting method for each time series according to its data characteristics. While Fildes and Petropoulos used a priori features from which to select the forecasters, other research has employed machine learning algorithms.

Prudêncio and Ludermir (2004) presented two case studies. In the first one, they defined ten features and two base forecasters, and used a C4.5 decision tree as the meta-learner to learn selection rules from 99 time series to identify when one base forecaster performs better than the remainder. Their results indicated that using the method recommended by the meta-learner on average provided more accurate forecasts than using either of the base forecasters as the default model. In the second case, they used a different meta-learner, named NOEMON which had been introduced by Kalousis and Theoharis

(1999), to rank three base forecasters. Wang, Smith-Miles, and Hyndman (2009) used a more comprehensive set of features, and, using both supervised and unsupervised meta-learners, provided selection rules, as well as visualizations in the feature space. They found that a derived weighting schema based on the rule induction that combined base forecasters could further improve forecasting accuracy over only using one of the base forecasters. Similarly, Lemke and Gabrys (2010) showed the superiority of a ranking-based combination with base forecasters over an aggregate model selection approach. Widodo and Budi (2013) proposed reducing the dimensionality of the feature set extracted from the time series by Principal Component Analysis before they are used for meta-learning. They found that the dimensionality reduction might help the meta-leaner to find the appropriate method. Kück, Crone, and Freitag (2016) proposed a meta-learner based on neural networks. They introduced a new set of features based on the forecasting errors of a set of commonly used forecasting methods and showed promising results when including error-based feature sets into the meta-learner for selecting between forecasters. Cerqueira, Torgo, Pinto, and Soares (2017) proposed a meta-learner to predict the absolute error of each of the base-forecasters, and then used the predicted error as the performance measure to ensemble base- forecasters.

Recently, Talagala, Hyndman, and Athanasopoulos (2018) proposed a meta learning framework named FFORMS (Feature-based FORecast Model Selection) that uses Random Forest as the meta-learner based on a set of 25 features for non-seasonal data and 30 features for seasonal data, to select the best single forecasting method from nine base forecasters. To build a reliable classifier, they proposed augmenting the set of observed time series by simulating new time series similar to those in the assumed population. Montero-Manso, Athanasopoulos, Hyndman, and Talagala (2018) built on FFORMS by using meta-learning to select the weights for a weighted forecast combination. Forecasts from all base forecasters were combined, and the weights used in the combination were chosen based on the features of each time series. They called this framework FFORMA (Feature-based FORecast Model Averaging). FFORMA resulted in the second most accurate point forecasts and prediction intervals amongst all competitors in the M4 competition (Makridakis & Petropoulos, 2019). One of the drawbacks of FFORMA is it targets minimizing the loss of combined errors of base-forecasters, not the loss from the combined forecasts directly, so it can result in suboptimal combinations. Table 1 provides a summary for a clear comparison of the research described here contrasting with earlier related works on meta-learning.

As is shown in Table 1, the meta-learning framework proposed in this research contributes to this stream of literatures mainly in four aspects:

(1) The proposed meta-learner targets identifying the best ensemble weights to combine the forecasts of base-forecasters to *minimize the error loss directly*.

(2) Existing meta-learning methods depend totally on unsupervised judgmentally selected features. However, it is a difficult task to determine useful features only judgmentally in order to capture

intrinsic properties embedded in various time series data when the feature space is large. The workload from the calculation of these features is also substantial. To that end, in this research, inspired by the deep feature learning for image classification (Bengio, Courville, & Vincent, 2013; LeCun & Bengio, 1995; LeCun, Kavukcuoglu, & Farabet, 2010), we design a meta-learning framework based on convolutional neural networks which can learn a supervised feature representation from raw multiple time series automatically.

Table 1 Research studies on time series forecasting with meta learning

Paper	Number of Time series		Feature extracting		Base forecaster		Meta learner	
	Training	Test	Approach of the feature extraction	Feature extracting from influential factors	Number of individual forecasting models	Number of pooled forecasting models	Model	Target
Prudêncio and Ludermir (2004)	99	99	Judgmental/ unsupervised	No	2-3	None	DT/MLP	Best forecaster/ ranking
Wang et al. (2009)	315	315	Judgmental / unsupervised	No	4	None	SOM /DT	Best forecaster
Lemke and Gabrys (2010)	222	222	Judgmental / unsupervised	No	4	None	DT/SVM/FNN	Best forecaster / ranking
Widodo and Budi (2013)	3003	1001	Judgmental / unsupervised	No	4	None	KNN	Best forecaster
Kück et al. (2016)	78	33	Judgmental / unsupervised	No	4	None	MLP	Best forecaster
Cerqueira et al. (2017)	14	14	Judgmental / unsupervised	No	9	None	RF	Predict the absolute error
Talagala et al. (2018)	1001/ 3003	3003 / 1001	Judgmental / unsupervised	No	9	None	RF	Best forecaster
Montero-Manso et al. (2018)	100,000	100,000	Judgmental / unsupervised	No	8	None	GBRT	Minimum loss of combined errors
This study	83944	36194	Automatic/ supervised	Yes	9	8	DCCNN	Minimum loss of combined forecasts

Abbreviation of forecasting models in the table. DT: Decision Trees; MLP: Multi-Layered Perceptron Neural Network; SOM: Self-Organizing Maps; SVM: Support Vector Machine; FNN: Feedforward Neural Network; KNN: K-Nearest Neighbour; RF: Random Forest; GBRT: Gradient Boosting Regression Trees; DCCNN: Double Channel Convolutional Neural Networks.

(3) We learn features from both sales time series and their influential factors, in contrast with the existing studies which have focused only on univariate time series. In the context of SKU level sales forecasting in retail, sales are affected significantly by a number of factors, such as price reduction, display, feature advertising, and special events. These factors are usually known in advance and can cover the whole forecasting horizon. Some earlier empirical research has shown that the performance of forecasting methods may be related to many influential factors. For example Gür Ali et al. (2009) and Ramos and Fildes (2017) found that while simple time series techniques performed well for periods without focal product promotions, for periods with promotions, methods including promotional drivers improved accuracy substantially. Fildes, et al. (2020) have summarized the research.

(4) The modeling strategies for forecasting many related time series can be classified into two categories: modeling each time series individually or modeling the group of time series together in a pooled fashion. Individual modelling can consider each time series' own characteristics, such as seasonality, trend and promotional elasticities, but is inefficient as it fails to capture any cross-sectional common patterns. Also as the data is often limited at individual series level, modeling each SKU individually may lead to noisy and often nonsensical estimates of the series specific elasticities (Blattberg & George, 1991). Pooled modeling can enhance the relevant data availability and this has the potential to capture cross time series common patterns, thereby improving the robustness of the estimated parameters (Dekker, van Donselaar, & Ouwehand, 2004; Zotteri & Kalchschmidt, 2007). But as price and promotional elasticities potentially vary considerably among chains and brands, one overall model may be overly restrictive in the light of each SKU's unique characteristics. While existing researches have so far used only individual forecasters as the base, we propose to combine the two modelling strategies to build a mixed base which is constituted of both individual and pooled forecasters, thereby taking advantage of both strategies in the meta learning.

3. Methodology

3.1. Problem formulation

In this research, we aim to forecast the SKU×Store level sales from time T+1 to T+H, given the data until time T. In addition to historical sales data, we also incorporate influential factors such as price, promotions, seasonality, and calendar events. We denote those influential factors for SKU i at time interval t as a vector $\mathbf{x}_{it}$, the sales history as $\left[y_{it} \right]_{t=1:\mathrm{T}}$, the goal being to predict

$$\left[\hat{y}_{i,T+h} \right]_{h=1:H} = f\left(\left[y_{it} \right]_{t=1:T}, \left[\mathbf{x}_{it} \right]_{t=1:(T+H)} \right), \tag{1}$$

where $f(\cdot)$ is the prediction function. In this research, the prediction function for each sales series is a weighted combination of a pool of base-forecasters, and the weights are obtained with a meta-learning algorithm.

3.2. Overview of the meta-learning framework

We propose a meta-learning framework with automatic feature learning for retail product sales forecasting, which is presented in Fig.1. Implementing the framework consists of two phases: meta-learning and meta-forecasting.

In the meta-learning phase, we need to first extract a large sub-set of sales time series and the corresponding history of influential factors from the historical database. Those extracted sales series should be similar to those we will be forecasting (same SKU, same category, or sold in the same stores). Large retailers have accumulated huge amounts of SKU level historical data, but because of the rapidly changing assortments, many SKUs have limited sales history at store level. We therefore do not assume that the sample of SKUs in the training set is the same as that in the test set. We only assume that the SKUs in both sets are forecasted with a rolling window of the same width, so that we can fit base-forecasters using the same length of data during the meta-learning and meta-forecasting phases.

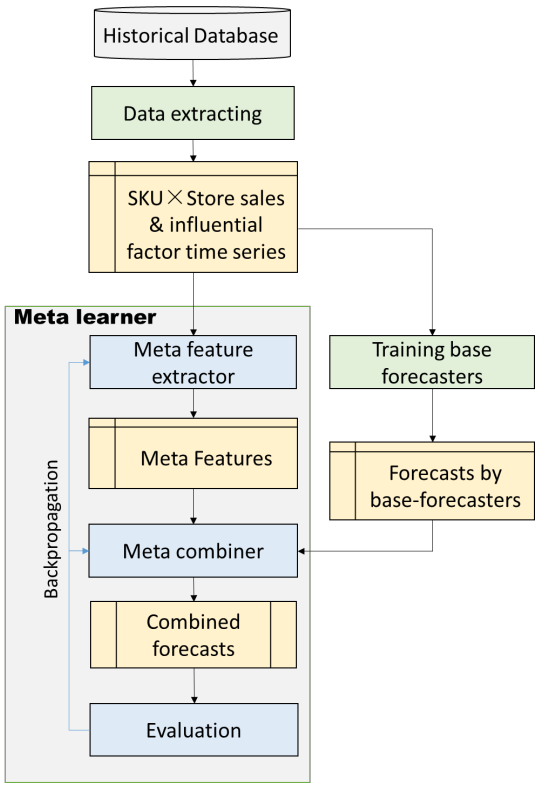

Figure 1. A meta-learning framework for retail sales forecasting

For each rolling period, we first fit a pool of base-forecasters, then generate *H*-step ahead forecasts

with the fitted models. The proposed meta-learner is an integration of two modules of neural networks (Fig.2). The raw time series that are used to fit the base-forecasters, are first fed into a module of convolutional neural networks, which is used to extract features from the time series data automatically. The extracted features are then fed into another module of the neural network to transform the features into a set of weights to combine the forecasts produced by the base-forecasters. The combined forecasts are further evaluated using the chosen loss function and then the network weights are updated through backpropagation algorithms.

During the meta-forecasting phase, we also need first to fit the same pool of base-forecasters for each sales time series to be forecasted, and to generate forecasts with the fitted models. Then those forecasts together with the raw time series are fed into the trained meta-learner. The trained meta-learner extracts the features from the time series, calculates combination weights, and generate a set of ensemble forecasts for each sales time series being forecast.

3.3. The structure of the meta-learner

Feature learning (or representation learning) has become an important field in the machine learning community in recent years (Bengio et al., 2013). The most successful feature learning framework adopts deep neural networks, which build hierarchical representations from raw data (LeCun & Bengio, 1995; LeCun et al., 2010; Lee, Grosse, Ranganath, & Ng, 2009). Particularly, one deep networks, Convolutional Neural Networks (CNN) can automatically mine and generate deep features of input images or time series, and has shown a strong robustness against data translation, scaling and rotation, this strength deriving from three important ideas that differ from traditional forward neural networks; they are as follows: local receptive field, weights sharing and pooling (LeCun & Bengio, 1995).

The convolutional neural network (CNN) has shown excellent performance in many computer vision, machine learning and pattern recognition problems, and especially, problems concerning feature learning from sequential data such as semantic role labelling (Santos & Zadrozny, 2014), sentence classification (Kim, 2014), machine translation (Kalchbrenner et al., 2016), audio synthesis (Oord et al., 2016), and time series classification (Zębik, Korytkowski, Angryk, & Scherer, 2017; Zheng, Liu, Chen, Ge, & Zhao, 2014). These earlier research works have highlighted the potential of CNNs showing better performance than traditional algorithms, and these findings have motivated us to investigate the feasibility of using feature learning in the meta-learning time series field.

We propose a novel meta-learner based on a Double Channel Convolutional Neural Network (DCCNN), which is shown in Fig.2. One channel takes the sales time series $\left[y_{i,t} \right]_{t=1:T}$ as the input and the other inputs multivariate time series of influential factors $\left[\mathbf{x}_{it} \right]_{t=1:T+H}$ simultaneously. More formally, for a d dimensional multivariate time series input (or the output of the preceding layer) $\mathbf{z} \in \mathbb{R}^{d \times T}$, and

a 1-D filter $v \in \mathbb{R}^s$ with stride *s*, the convolution operation *C* on time step *t* of the input **z** is defined as

$$C(t) = \left(\mathbf{v} * \mathbf{z}\right)_t = \sum_{j=1}^{d} \sum_{i=0}^{s-1} v_i \mathbf{z}_{j,t-i} \,. \tag{2}$$

If we have *K* filters, we can then write the outputs of a convolution layer as $\mathbf{U} = \left[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_K\right]$, where $\mathbf{u}_k = \left[\left(\mathbf{v}_k * \mathbf{z}\right)_t \right]_{t=s:T}$.

In the proposed DCCNN as shown in Fig.2, both convolutional channels contain three stacked temporal convolutional blocks, used as a feature extractor. Each convolutional block contains a convolutional layer and a ReLU activation ($ReLU\left(u\right) = \max(0, u)$).The first two convolutional blocks conclude with a squeeze and excite layer (Hu, Shen, Albanie, Sun, & Wu, 2019). The squeeze operation exploits the contextual information outside each filter's focused feature of the input time series by using a global average pool to generate summary statistics over the learnt feature map. Specifically, the convolution layer output, **U**, is shrunk through temporal dimensions *T* to compute the summary statistics, $\overline{\mathbf{U}}$. The *k*-th element of $\overline{\mathbf{U}}$ is calculated by

$$\overline{\mathbf{u}}_k = \frac{1}{T} \sum_{t=1}^{T} \mathbf{u}_{k,t} \,. \tag{3}$$

The summary information from the squeeze operation is followed by an excite operation, whose objective is to capture the dependencies among the learnt features. To achieve this, a simple gating mechanism is applied with a sigmoid activation, as follows:

$$\mathbf{q} = \sigma\left(\mathbf{w}_2 \,\mathrm{ReLU}\left(\mathbf{w}_1 \overline{\mathbf{U}}\right)\right), \tag{4}$$

where σ is a sigmoid activation, $\mathbf{w}_1 \in \mathbb{R}^{\frac{K}{r} \times K}$ and $w_2 \in \mathbb{R}^{K \times \frac{K}{r}}$ are learnable weights. $\mathbf{w}_1$ are the parameters of the dimensionality-reduction layer and $\mathbf{w}_2$ are the parameters of the dimensionality increasing layer. *r* is a dimensionality reduction ratio of the gating mechanism which can dynamically control the information flow based on the current input. Finally, the output of the block is rescaled as follows:

$$\tilde{\mathbf{u}}_k = q_k \cdot \mathbf{u}_k \,, \tag{5}$$

The excitation operator maps the summary statistics $\overline{\mathbf{U}}$ to a set of weights. In this regard, the squeeze and excite block intrinsically introduces dynamics conditioned on the feature map **U**, which can be regarded as a self-attention function[2] to allow placing more weights on the relevant features as needed. Hu et al. (2019) showed that the squeeze and excite block can improve the quality of representations produced by the convolutional layer.

[2] See Vaswani et al. (2017) for more information on attention mechanisms.

The last temporal convolutional block is followed by a global average pooling layer, which is used to reduce the number of parameters in the network. The outputs of the global pooling layer from both channels are concatenated and then fed into a dropout layer to mitigate overfitting (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). We then use a dense layer with a softmax activation to transform the learnt features into a set of weights whose dimension equals to the number of base-forecasters.

Simultaneously, the H step ahead forecasts $\left[\hat{y}_{i,T+h}^{(m)}\right]$, $h=1:H$, $m=1:M$, of the sales time series i given by the M base-forecasters are inputted. Using the weights obtaining from the dense layer with softmax activation, the M forecasts are weighted and summed to generate a set of combination forecasts,

$$\hat{y}_{i,T+h}=\sum_{m} w_i^{(m)}\hat{y}_{i,T+h}^{(m)} \quad , \tag{6}$$

where $\hat{y}_{i,t+h}^{(m)}$ is the forecasts generated from m^{th} base-forecaster and w are ensemble weights from softmax layer and $\sum_{m} w_i^{(m)}=1$. The combination forecasts are evaluated using a Scaled Mean Square Error (SMSE) loss function, which is defined as

$$L(\theta)=\frac{1}{N}\sum_{i=1}^{N}\frac{\frac{1}{H}\sum_{h=1}^{H}(\hat{y}_{i,T+h}(\theta)-y_{i,T+h})^2}{S_i}, \tag{7}$$

where θ is the set of all parameters to be estimated in the network， and S_i is the averaged MSE of the M base forecasters, that is defined as

$$S_i=\frac{1}{MH}\sum_{m=1}^{M}\sum_{h=1}^{H}(\hat{y}_{i,T+h}^{(m)}-y_{i,T+h})^2 . \tag{8}$$

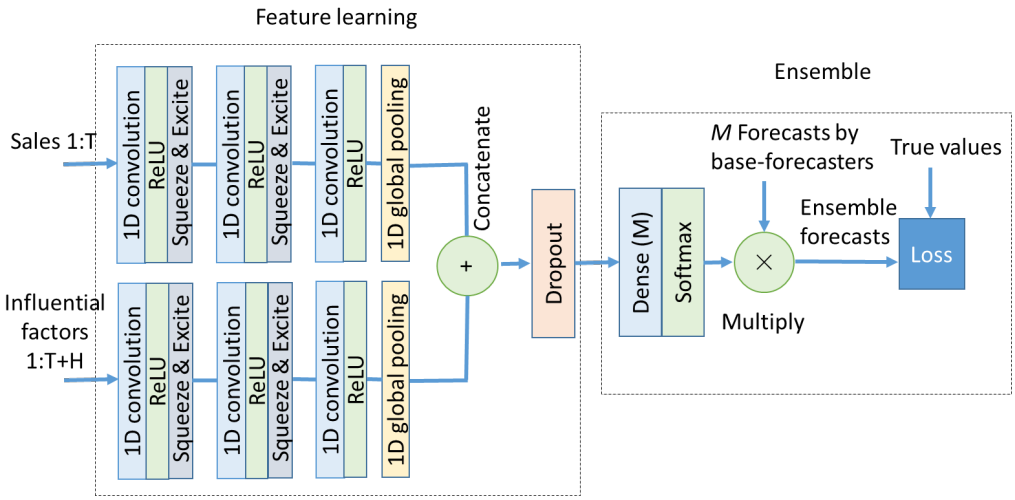

Figure 2 The network structure of the meta-learner

3.4. Base forecasters

To collect a pool of base forecasters for the proposed meta-learner, we consider forecasting methods based on two complementary modeling strategies: modeling each sales series separately, leading to a heterogeneous set of models and modeling all sales series in the study together resulting in a pooled homogenous model.

We consider five forecasting methods under the individual modeling strategy. These methods are explained as follows: Most of them have been considered and shown promising forecasting performance in the retail sales forecasting literatures.

(1) ExponenTial Smoothing (ETS) state space model. ETS is a univariate forecasting model using only the past sales history, and has been employed as benchmark in a number of researches in retail forecasting (e.g.,Ma, Fildes, & Huang, 2016; Ramos & Fildes, 2017). Those researchers have found that ETS performed well for periods with low promotional intensity or for products with a low price elasticity of demand.

(2) Autoregressive Distributed Lag (ADL) model. ADL is a multiple linear regression model in nature whose exogenous inputs correspond to the lags of sales, calendar events, price reduction, and promotions. Huang, Fildes, and Soopramanien (2014) and Ma et al. (2016) evaluated ADL models on SKU level sales data in a number of stores. They found that ADL on average outperforms the univariate methods with gains typically above 10%. For promotional periods the gains are typically higher.

(3) Autoregressive integrated moving average with external variables (ARIMAX) model. Compared with ADL, ARIMAX is more sophisticated as it includes complex error correlation structures though typically fewer variables in the feature set. Arunraj and Ahrens (2015) used an ARIMAX model to forecast the daily sales of bananas in a German retail store, and showed that the ARIMAX model outperformed an ARIMA model and two neural network models.

(4) Support Vector Regression (SVR). Support vector regression is an artificial intelligence forecasting tool based on statistical learning theory and a structural risk minimization principle (Vapnik, 1999). Lu (2014) used the SVR as a benchmark for sales forecasting of computer products, but he did not find it provided superior performance on average over univariate time series models. Here SVR is considered as one of the base-forecasters as its increased generality might improve the potential pool to provide more accurate forecasts for some of the sales series under study.

(5) Extreme learning machines (ELM). The ELM is a learning algorithm for single-hidden-layer feedforward Neural Networks. It has been adopted in a number of fashion retail forecasting studies (Wong & Guo, 2010; Xia, Zhang, Weng, & Ye, 2012; Yu, Choi, & Hui, 2011). The experimental results have shown that the performance of the ELM is more effective than traditional Back Propagation Neural Networks (BPNN) models for fashion sales forecasting, though their accuracy in practice compared to BPNN is at best moot.

Under the pooled modeling strategy, we consider four forecasting methods:

(1) ADL with data Pooling (ADLP). Pooled regression is a practical approach to forecasting product level sales in retail. Andrews, Currim, Leeflang, and Lim (2008) found that accommodating store-level heterogeneity did not improve the accuracy of marketing mix elasticities relative to the homogeneous model, and the improvements in fit and forecasting accuracy were also modest. Gür Ali et al. (2009) also found that pooling observations across stores and subcategories provided better predictions than pooling across either only stores or only subcategories. Based on those empirical results, we adopt a homogeneous model that pools SKUs across stores and categories.

(2) ELM with data Pooling (ELMP). Similar with ADLP, ELM (Extreme learning machines) can also be used to train a homogeneous model by pooling SKUs across stores and categories. Compared with individual ELMs, we consider a larger number of hidden neurons in ELMP compared to that of ELM, thereby increasing model complexity in order to adapt to the richer data condition arising from using the pooled data set.

(3) Random Forest (RF). Random Forest is based on decision trees combined with aggregation and bootstrap ideas and was first proposed by Breiman (2001). The data are split in such a way as to train a large number of decision tree models separately with forecasts produced from each sub-model, then combined. Random Forest has widely been used in applications, see Ziegler and König (2014) for a recent survey. Recently, Ma and Fildes (2020) adopted RF as a benchmark model in forecasting customer flows with mobile payment data and showed its superior performance over pooled regression.

(4) Gradient Boosting Regression Trees (GBRT). GBRT is one of most established gradient boosting algorithms, which uses a regression tree as the base weak learner (Friedman, 2001). GBRT has empirically proven itself to be highly effective for a vast array of classification, ranking and regression problems. It is one of the most preferred choices in data analytics competitions such as Kaggle and the KDD Cup and has also showed its potential in time series forecasting (Ma and Fildes, 2020).

4. Experimental design

In this section we present the experiments carried out to test the performance of the proposed meta-learning framework in forecasting SKU level sales. These address the following research questions:

Q1: How does the forecasting performance of the proposed meta-learner compare to the performance of the base forecasters for the SKU level sales forecasting tasks? And to a simple combination of the forecasters? Whether there are circumstances when the meta-learner is particularly effective, e.g., for promoted periods?

Q2: How does the performance of this novel meta-learner compare to the performance of the FFORMA meta-learner?

Q3: How does the forecasting performance of the proposed supervised feature learning method compare to the performance of commonly used hand-selected features?

Q4: Is it beneficial to extract features from potentially influential factors in addition to the historical sales time series? And to the past accuracy statistics of the forecasters?

Q5: Is it beneficial to use a mixed pool of base-forecasters composing of forecasting methods using both individual and data pooling modelling strategies?

Q6: Is it beneficial to target finding the best ensemble forecasts with the meta-leaner instead of looking to identify the best individual forecaster?

Overall, by answering these research questions we aim to provide insight into how the new meta-learner performs compares to key benchmarks and the circumstances when it is particularly (in)effective.

4.1. Data

The empirical data comes from the IRI dataset (Bronnenberg, Kruger, & Mela, 2008)[3]. The IRI dataset includes grocery and drug chain data from a sample of stores in 50 markets and 30 categories, involving approximately 25%-30% of the consumer packaged goods sales in a grocery store. This is weekly data by SKU and includes information on sales, price, feature advertisings and displays. Based on the objectives of this research, to mimic a retail chain wide forecasting requirements, we have selected 6 product categories, 'milk', 'beer', 'mayo', 'yogurt', 'coffee' and 'laundet'(laundry detergent), and then randomly selected 100 stores which had sold these product for the last three years of the data set. These records of sales and promotions were then extracted for those categories and stores for the last 153 weeks.

We used a fixed rolling window with the width of 55 weeks for estimation and forecasting. We moved the window forward every 7 weeks to generate 15 slots of data over 153 weeks of the data sample (Figure 3).

[3] All estimates and analyses in this paper based on Information Resources, Inc. data are by the author and not by Information Resources, Inc.

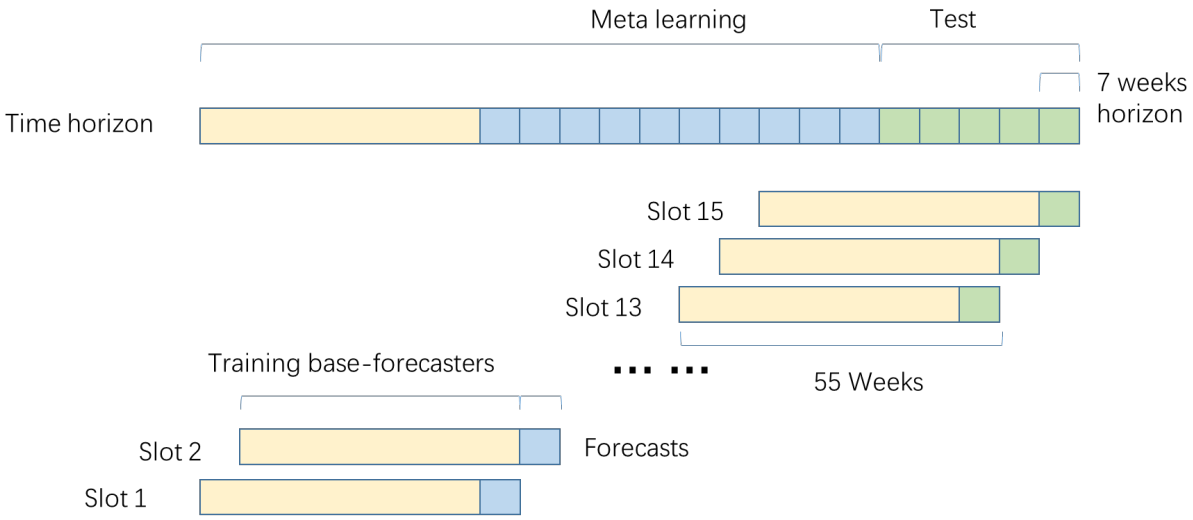

Figure 3 Data manipulation to generate training and test data

The first 10 slots were used as training data for meta-learning, and the last 5 slots used to test the forecasting performance of the meta-learner. For each slot of data, the first 48 weeks of data were used to train the base forecasters, and the last 7 weeks of data used to evaluate the performance of 12 base forecasters. In each slot, SKUs with discontinued sales were excluded in our experiments as we do not know whether the missing sales were due to stockout (the IRI dataset does not provide inventory information) or intermittent demand. Table 2 presents the means of units sold per week and percentages of weeks concerning promotional activities, including price reductions (more than 5 percent), displays and features in both training and test data slots.

Table 2 Description statistics of the data sample

Data	Num of Slots	Num of SKUs	Num of Sales time series	Mean units sold per week	Proportions of weeks concerning promotional activities		
					Price reductions	Displays	Features
Training	10	2944	83944	33.10	0.19	0.07	0.10
Test	5	2039	36194	22.64	0.20	0.08	0.11

4.2 Training base forecasters

For each slot of data, we in turn trained 9 base forecasters using the first 48 weeks of data in the slots, and then generated forecasts with each trained model for the remaining 7 weeks. In table 3, we summarize some details for the training process, including the explanatory variables used in the model, the software tool to implement the method and the settings for the hyper-parameters.

For ETS and ARIMAX, sales time series were used to train the model after a log transformation, and then the next 7 weeks of forecasts were generated recursively. To train the other base-forecasters, each SKU sales time series was transformed into a regression matrix and the dependent and

independent variables defined. We adopted a Direct strategy to generate multi-step ahead forecasts as it has been shown to be more robust than the recursive approach and is easy to implement, though it demands more computational time (Ma & Fildes, 2020). Specifically, the sales forecasting on SKU *i* by method *m* for horizon *h* for any given forecast origin is given by

$$\hat{y}_{i,T+h}^{(m)} = f_h^{(m)}\left(\left[y_{it}\right]_{t=(T-L):T}, \left[\mathbf{x}_{it}\right]_{t=(T-L):T+H}\right). \tag{9}$$

For horizon *h*, only sales at least *h* steps before the target time period can be used to construct explanatory features, but the external explanatory variables, including lags of price reduction, display, feature advertising and calendar events, do not have such a limitation as they are assumed to be known in advance or under control.

For base forecasters using individual forecasting, we estimated the model with 1 or 3 lags, i.e., L=1 or 3; for base forecasters using data pooling, L is set to be 3 or 7 to allow more complex homogeneous models. Finally, to account for the log transformation bias, for each base-forecaster, *m*, we multiply their forecasts by a bias adjustment factor $\alpha^{(m)}$, which is estimated with their respective forecasts in the training data to let

$$\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{h=T+1}^{T+H}\alpha^{(m)}\hat{y}_{ih}^{(m)} \Big/ \sum_{h=T+1}^{T+H} y_{ih}\right) = 1. \tag{10}$$

Table 3 Training settings for base forecasters

Base forecaster	Software tool	Hyper-parameters
1 ETS	'ets' function in the 'forecast' R package v.8.4 (Hyndman & Khandakar, 2008)	Under default settings
2 ADL	'glmnet' package v.2.0-16 in R (Friedman, Hastie, & Tibshirani, 2010)	The penalty parameter is determined by 10 folds cross-validation
3 ARIMAX	'auto.arima' function in the 'forecast' R package v.8.4 (Hyndman & Khandakar, 2008)	Under default settings
4 SVR	'e1071' R package v 1.7.2 (Meyer et al., 2019)	Using radial kernel under default settings
5 ELM	'elmNNRcpp' R package v 1.0.1 (Mouselimis & Gosso, 2018)	Using 5 hidden neurons and linear activation, all others are under default settings
6 ADLP	'glmnet' package v.2.0-16 in R (Friedman et al., 2010)	The penalty parameter is determined by 10 folds cross-validation
7 ELMP	'elmNNRcpp' R package v 1.0.1 (Mouselimis & Gosso, 2018)	Using N/100 hidden neurons (N is the number of sales series in the data slot) and linear activation, all others are under default settings
8 RF	'Ranger' package v.0.11.2 in R	Under default settings

	(Wright & Ziegler, 2017)	
9 GBRT	'Xgboost' package v. 0.82.1 in R (Chen & Guestrin, 2016)	Arbitrary choosing 0.02 as the learning rate, and 1000 as the rounds of training, all others are under default settings

4.3 Training meta-learners

Benchmark meta-learners

To investigate our research questions Q2 to Q6, based on the meta-learner introduced in the section 3.2, we designed a series of meta-learners as benchmarks. Table 4 provides a comparative summary of the proposed meta-learner with benchmarks. M0 represents the meta-learner introduced in section 3.2, and M1 to M6 are described in detail in the following.

Table 4 Comparative summary of meta-learners

Meta learner	Automatic feature learning	Extracting features from influential factors	Pool of base-forecasters	Target
M0	Yes	Yes	Mixed	Best ensemble
M1	No	Yes	Mixed	Best ensemble
M2	Yes	No	Mixed	Best ensemble
M3	Yes	Yes	Only methods using Individual forecasting	Best ensemble
M4	Yes	Yes	Only methods using Pooled forecasting	Best ensemble
M5	Yes	Yes	Mixed	Best forecaster
M6	Yes	Yes	Mixed	Predict absolute errors
FFORMA1	No	No	Mixed	Best ensemble
FFORMA2	No	Yes	Mixed	Best ensemble

(1) The meta-learner with unsupervised hand-selected features (M1).

We selected a set of hand-selected features describing the characteristics of weekly sales data in

our experiments which are listed in Table 5. All of the features have been previously used in Montero-Manso et al. (2018), and the functions to calculate these are implemented in the 'tsfeatures' R package by Hyndman et al. (2019). In contrast to Montero-Manso et al. (2018) we extracted features from the sales time series to be forecasted, and additionally, from the multivariate time series of influential factors.

As we no longer need to do feature learning when using hand-selected features, we designed a Fully Connected Neural Networks (FCNN) instead of DCCNN as meta-learner, which is shown in Figure 4. FCNN took the hand-selected features as the input, and used three layers of a fully connected neural network with 128, 64 and 32 units and ReLU activations to process the inputs. Then a dropout layer with rate 0.8 was applied to mitigate overfitting. The remaining features were then fed into a dense layer with softmax activation, which transforms them into a set of weights. The following components were the same as in M0. We did not search for the optimal FCNN networks with an automatic algorithm. Instead we considered FCNNs with different numbers of hidden layers, i.e., from two to four, and compared their forecasting performances on the validation data. We found the performances were in general quite robust to these changes in NN specification. So we arbitrarily selected the three layer FCNN structure for M1 to conduct the experiments.

Table 5. Hand-selected features on Sales(S), Price reduction (P), and display & feature advertising (DF)

Feature	Description	S	P	DF
1 trend	strength of trend	√		
2 linearity	linearity	√		
3 curvature	Curvature	√		
4 spikiness	spikiness	√		
5 e_acf1	first ACF value of remainder series	√		
6 e_acf10	sum of squares of first 10 ACF values of remainder series	√		
7 stability	Stability	√	√	√
8 lumpiness	Lumpiness	√	√	√
9 entropy	spectral entropy	√	√	√
10 hurst	Hurst exponent	√	√	√
11 nonlinearity	Nonlinearity	√		
12 alpha	Alpha estimation in ETS(A,A,N)	√		
13 beta	Beta estimation in ETS(A,A,N)	√		
14 ur_pp	test statistic based on Phillips-Perron test	√		
15 ur_kpss	test statistic based on KPSS test	√		
16 y_acf1	first ACF value of the original series	√		
17 diff1y_acf1	first ACF value of the differenced series	√		
18 diff2y_acf1	first ACF value of the twice-differenced series	√		
19 y_acf10	sum of squares of first 10 ACF values of original series	√		
20 diff1y_acf10	sum of squares of first 10 ACF values of differenced series	√		
21 diff2y_acf10	sum of squares of first 10 ACF values of twice-differenced	√		
22 y_pacf5	sum of squares of first 5 PACF values of original series	√		
23 diff1y_pacf5	sum of squares of first 5 PACF values of differenced series	√		
24 diff2y_pacf5	sum of squares of first 5 PACF values of twice-differenced	√		

25 crossing_point	number of times the time series crosses the median	√	√	√
26 flat_spots	number of flat spots, calculated by discretizing the series	√	√	√
27 ARCH.LM	ARCH LM statistic	√		

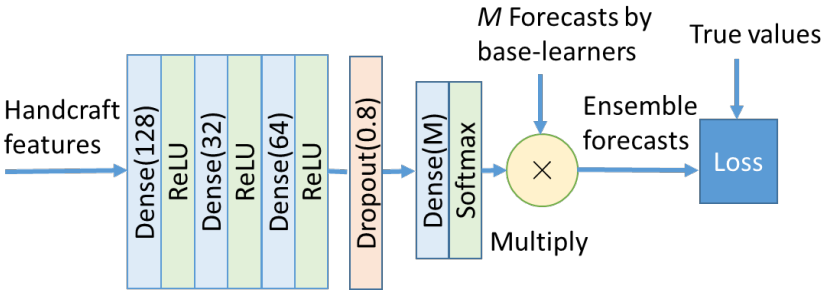

Figure 4 The network structure of meta-learner using hand-selected features

(2) The meta-learner without feature learning from influential factors (M2)

This meta-learner is similar with the DCCNN shown in Figure 2, but used only the first channel to learn features from just the sales time series.

(3) The meta-learner with base-forecasters that only used the individual modeling strategy (M3)

This meta-learner has the same structure with the DCCNN shown in Figure 2, but using a limited pool of base-forecasters composing of only methods modeling and forecasting each SKU individually.

(4) The meta-learner with base-forecasters only using pooled modeling strategy (M4)

This meta-learner also has the same structure as the DCCNN shown in Figure 2, but using a limited pool of base-forecasters composing only methods modeling all SKUs in each of the data slots (Fig.3) in a pooled fashion.

(5) The meta-learner targets identifying the best individual base-forecaster (M5).

This meta-learner also uses DCCNN to extract features from sales and influential factors, and feeds these learned features into a softmax layer. But the outputs from the softmax are interpreted as the probabilities that each of the base-forecasters provides the most accurate forecasts for the current sales series, and is therefore a classification problem in this meta-learner (Figure 5). The loss function here is categorical cross-entropy which is defined as

$$L(\theta) = -\sum_{i=1}^{N}\sum_{m=1}^{M} l_i^{(m)} \log\left(w_i^{(m)}\right), \tag{11}$$

where $w_i^{(m)}$ is outputted from the softmax layer indicating the probability that the m^{th} base-forecaster is the best method to forecast SKU i, and $l_i^{(m)}$ is the true label of either 1 or 0, indicating the m^{th} base-forecaster performs the best or not on SKU i. The label is evaluated by mean absolute error in the seven weeks of the forecasting horizon across base-forecasters. During the meta forecasting phase, the meta-learner selects the base-forecaster which has maximum probability to be the best and this is then used to generate forecasts.

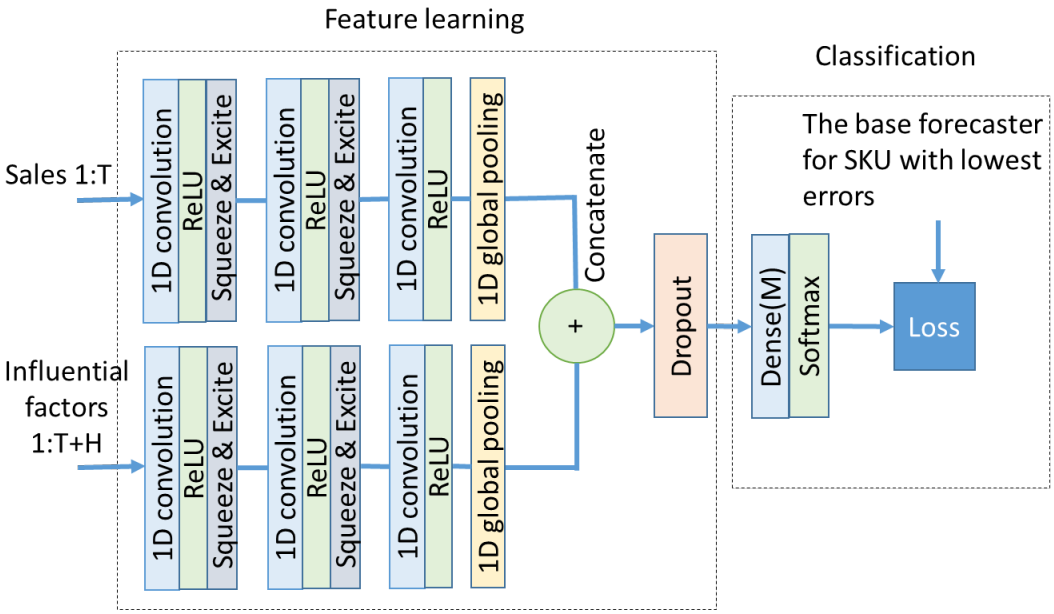

Figure 5 The network structure of the meta-learner targeting identifying the best performing base forecaster

(6) The meta-learner predicts base-forecasters' absolute errors (M6)

This meta-learner is adopted from Cerqueira et al. (2017), which aims to predict the absolute forecasting errors arising from each base-forecaster. The structure of this meta-learner is illustrated in Figure 6.

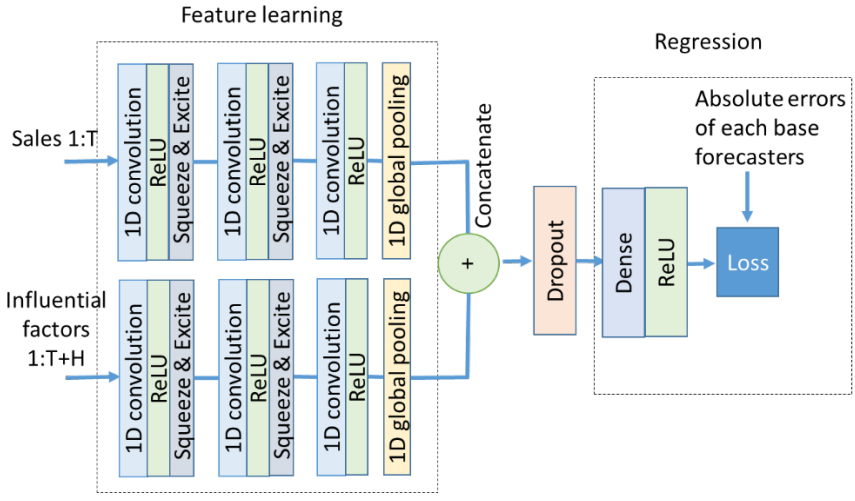

Figure 6 The network structure of the meta-learner targeting at predicting forecasting error loss function

The loss function here is defined as

$$L(\theta)=\frac{1}{NM}\sum_{i=1}^{N}\sum_{m=1}^{M}(\hat{e}_i^{(m)}-e_i^{(m)})^2\,, \tag{12}$$

where $e_i^{(m)}=\sum_{h=1}^{H}\left|\hat{y}_{i,T+h}^{(m)}-y_{i,T+h}^{(m)}\right|$ represents the absolute errors generated by m^{th} base-forecaster on forecasting SKU i, and $\hat{e}_i^{(m)}$ is the output from the ReLU layer and represents the predicted error. During the Meta forecasting phase, the trained meta-learner generates absolute error predictions for each of the base-forecasters, and these are then used as weights in the ensemble of the base-forecasters according to their predicted performance:

$$\hat{y}_{i,T+h}=\sum_m w_i^{(m)}\hat{y}_{i,T+h}^{(m)},\ \ w_i^{(m)}=\frac{\exp\left(-\hat{e}_i^{(m)}\right)}{\sum_m \exp\left(-\hat{e}_i^{(m)}\right)}. \tag{13}$$

(7) FFORMA

FFORMA uses the gradient tree boosting model of xgboost (Chen & Guestrin, 2016) as the underlying implementation of the learning model (Montero-Manso et al., 2018). The original version of the FFORMA was developed for pure time series forecasting tasks, so it considered eight univariate time series methods as the base forecasters and adapted the Overall Weighted Average (OWA) error as the measure for forecasting loss, which adds together the Mean Absolute Scaled Error and the symmetric Mean Absolute Percentage Error. For a fair comparison, FFORMA as implemented in this research used the same pool of base forecasters as that in M0, and also used the SMSE as the loss measure. To train the xgboost, we adopted a set of hypermeters which had been optimized for the M4 time series competition from the source code of FFORMA at github.com/robjhyndman/M4metalearning. To investigate the effects of using influential factors on the performance of FFORMA, we tested two versions of the model: the first is named as FFORMA1 which uses only features extracted from the sales series, and the other, FFORMA2, uses additional features extracted from influential factors (as with M1).

Preprocessing and training process

For all the meta-learners, we normalized the time series (hand-selected features for M1) by using z-normalization before inputting. To estimate the parameters of the neural networks, we utilized a gradient based optimization method to minimize the loss function. We used Keras with Tensorflow as the backend to implement our proposed model, and used Adam (Kingma & Ba, 2015) for optimization.

All the experiments were run on a workstation with one NVIDIA Titan XP GPU. Neural networks are inherently parallel algorithms, and GPUs can take advantage of this parallelism to accelerate the training process. In our experiments, the time for each training epoch was between ten to thirty seconds, depending on the network structure of the meta learners. The batch size for training all the meta-leaners

in our experiment was set to be 4096. The first eight slots of the training samples were selected for training each model and the remaining 2 slots are in the validation set for parameter tuning. We find that 50 epochs are enough for all the meta learners to reach the minimum from our tuning results. The filters of three block of the convolution networks in the meta-learners are set to be 64, 128 and 64 respectively and the dropout rate was set to be 0.8. In addition, we utilize the initialization proposed by He, Zhang, Ren, and Sun (2015) for all convolutional layers.

4.4. Combination benchmarks

We also employ three widely used combination approaches as the benchmarks.

(1) Equivalent weights combination (E1). All base forecasters are simply averaged using the arithmetic mean. It is popular due to its ease of implementation, robustness, and good record in economic and business forecasting (Barrow & Kourentzes, 2016).

(2) A weighted linear combination (E2). The weights are calculated according to their performance on the training data with a softmax function (Cerqueira et al., 2017).

(3) Equivalent weights combination over selected base forecasters (E3). Instead of using all the base forecasters in the E1, we select the four best base forecasters according to their performance on the training data, and then combine those selected forecasters using the equivalent weights method. This combiner is employed due to the suggestion of one of the referees.

OLS and constrained regression weights were also examined but performed poorly.

4.5. Forecasting evaluation metrics and validation

We use three error measures to compare the forecasting performance of the models. The first is sMAPE, symmetric Mean Absolute Percentage Error, defined for each forecast origin, which measures the difference between the prediction and the actual outcome, and is here defined as:

$$\text{sMAPE} = \frac{1}{NH}\sum_{i=1}^{N}\sum_{h=T+1}^{T+H}\left|\frac{\hat{y}_{ih}-y_{ih}}{\hat{y}_{ih}+y_{ih}}\right|, \tag{14}$$

where $\hat{y}_{ih}$ is the forecasts of SKU i in horizon h, and y_{ih} is the observed sales of SKU i in week h, N is the number of SKUs in the sample. It was chosen despite its known weaknesses because of its use in the M4 competition. The average Relative Mean Absolute Error (AvgRelMAE) is the second metric, which is proposed by Davydenko and Fildes (2013) for measuring forecasting accuracy at a disaggregate level (e.g. Store demand, SKU-level demand). It is a geometric mean of the ratio of the MAE between the candidate model and the benchmark model.

$$AvgRelMAE = \left(\prod_{i=1}^{N} \frac{\sum_{h=T+1}^{T+H} \left| \hat{y}_{ih} - y_{ih} \right|}{\sum_{h=T+1}^{T+H} \left| \hat{y}_{ih}^{0} - y_{ih} \right|} \right)^{\frac{1}{N}} \quad (15)$$

where $\hat{y}_{ih}^{0}$ is the baseline statistical forecast for SKU i, $\hat{y}_{ih}$ is the candidate model evaluated for SKU i. It has the advantage over sMAPE as removing outliers giving a more normal distribution and being readily interpretable.

In order to measure the forecasting error bias, the Mean Percentage Error (MPE), that is defined here as the mean of the ratios of total error to total sales in the test periods per SKU, accumulated over the forecast horizon, i.e.,

$$\text{MPE} = \frac{100}{N} \sum_{i=1}^{N} \left(\frac{\sum_{h=T+1}^{T+H} \left(\hat{y}_{ih} - y_{ih} \right)}{\sum_{h=T+1}^{T+H} y_{ih}} \right). \quad (16)$$

is used as the final criterion.

All these error measures are calculate for each forecast origin and then averaged over origins (in training or test data sets).

In addition, to evaluate whether any forecast accuracy differences in methods that may appear are due to randomness, we employ the non-parametric Friedman test and the post-hoc Nemenyi test (Demšar, 2006; Koning, Franses, Hibon, & Stekler, 2005). We use the implementation of the tests available in the 'tsutils' R package (Kourentzes, 2019).

The forecasting methods employed here are complex and even relatively simple procedures have often shown themselves to be not reproducible (Boylan, Goodwin, Mohammadipour, & Syntetos, 2015). Although the individual modules have been validated by being based on established R libraries, the meta-learning schema as shown in Fig. 1 is novel. The code has been made available on github.com/Shawn-nau/retail-sales-forecasting-with-meta-learning to allow others to check replicability. The IRI dataset as noted can be accessed through Information Resources Inc. Finally, the results we report we claim have face validity.

5. Results

5.1. Forecasting performance of the base-forecasters

The forecasting performance of all the base-forecasters on both training and test periods are shown in the Table 6 and Table 7 respectively. The ETS forecasts are used as the baseline for calculating AvgRelMAE. The results are very similar whichever of the two error measures are used. The results are also similar whether focusing on the training or test data sets. In both training and test periods, base-

forecasters which model each SKU series individually perform worse than those trained with pooled data. The GBRT and random forest showed similar results (a 16% improvement over ETS), and both models provide superior accuracy than all other base-forecasters on both data sets and both accuracy metrics. Using median measures provided similar results qualitatively.

Among individual forecasting models, models that used one lag have higher accuracy than the same model using three lags. On the contrary, models based on the pooled data using seven lags have better performance than the same model using three lags. The results also show that machine learning models, i.e., SVR and ELM, under the individual modeling strategy perform worse than simple ADL regressions. Under the data pooling strategy however, machine learning methods perform better than pooled regressions. The results provide further evidence that machine learning methods can provide more accurate forecasts than simple linear forecasting methods only when using data pooling. This implies that for forecasting methods using the individual modeling strategy it is better to keep the methods as simple as possible to avoid overfitting. Methods using the data pooling strategy should increase the model specification complexity to avoid underfitting.

The bias adjustment factors for each base-forecasters are shown in last column in Table 6. Table 7 shows that the adjustments work well, the maximum bias among all the base-forecasters is only around 2 percent.

Table 6 Forecasting performance of base-forecasters in training set

Base forecaster	Horizon								
	h=1		h=4		h=7		h=1-7		
	sMAPE	AvgRel MAE	sMAPE	AvgRel MAE	sMAPE	AvgRel MAE	sMAPE	AvgRel MAE	Bias adj.
ETS	19.367	1.000	20.366	1.000	20.859	1.000	20.137	1.000	1.043
ADL-1	16.717	0.871	17.516	0.862	17.672	0.840	17.200	0.855	1.017
ADL-3	16.898	0.878	17.724	0.870	17.944	0.854	17.417	0.864	1.019
ARX-1	17.198	0.897	17.976	0.885	18.210	0.874	17.716	0.884	0.997
ARX-3	18.074	0.941	18.828	0.930	18.982	0.911	18.529	0.922	0.987
ELM-1	18.142	0.952	19.441	0.981	19.356	0.931	18.902	0.944	0.980
ELM-3	19.705	1.043	20.679	1.037	20.770	0.999	20.337	1.026	1.015
SVM-1	17.164	0.912	17.812	0.897	17.915	0.870	17.534	0.886	1.001
SVM-3	17.509	0.926	18.213	0.910	18.427	0.895	17.964	0.908	1.012
GBRT-3	16.231	0.844	17.097	0.845	17.304	0.831	16.785	0.841	1.029
GBRT-7	16.144	0.842	16.930	0.838	17.320	0.831	16.709	0.839	1.021
ADLP-3	16.727	0.876	17.569	0.878	17.675	0.853	17.230	0.869	1.033
ADLP-7	16.593	0.869	17.414	0.869	17.667	0.853	17.139	0.865	1.031
RF-3	16.304	0.842	17.108	0.842	17.328	0.826	16.815	0.839	1.036
RF-7	16.235	0.837	16.985	0.834	17.365	0.828	16.762	0.835	1.040

ELMP-3	16.614	0.866	17.532	0.875	17.638	0.853	17.173	0.867	1.033
ELMP-7	16.496	0.858	17.345	0.863	17.670	0.851	17.090	0.862	1.035

The top two performed models are shaded; The ETS forecasts are used as the baseline for calculating AvgRelMAE.

Table 7 Forecasting performance of base-forecasters in test set

Base forecaster	Horizon								
	h=1		h=4		h=7		h=1-7		
	sMAPE	AvgRel MAE	sMAPE	AvgRel MAE	sMAPE	AvgRel MAE	sMAPE	AvgRel MAE	MPE
ETS	19.305	1.000	20.105	1.000	21.152	1.000	20.219	1.000	2.163
ADL-1	16.842	0.885	17.691	0.871	18.743	0.867	17.756	0.869	-0.339
ADL-3	16.999	0.893	17.929	0.883	19.053	0.886	17.957	0.882	0.247
ARX-1	17.246	0.907	18.059	0.904	18.996	0.894	18.190	0.902	1.045
ARX-3	18.101	0.952	18.924	0.945	19.900	0.942	19.055	0.947	1.154
ELM-1	17.959	0.932	19.280	0.956	20.629	0.984	19.448	0.970	0.254
ELM-3	19.744	1.048	20.679	1.043	21.654	1.038	20.739	1.041	1.602
SVM-1	17.175	0.915	17.950	0.909	18.833	0.894	18.058	0.907	1.283
SVM-3	17.531	0.925	18.380	0.922	19.347	0.920	18.467	0.923	1.693
GBRT-3	16.372	0.846	17.353	0.853	18.601	0.861	17.379	0.847	-1.653
GBRT-7	16.201	0.844	17.137	0.842	18.597	0.870	17.301	0.847	-1.558
ADLP-3	16.788	0.869	17.815	0.880	18.902	0.879	17.805	0.872	-1.927
ADLP-7	16.673	0.863	17.608	0.868	18.868	0.881	17.692	0.867	-1.514
RF-3	16.454	0.848	17.351	0.846	18.580	0.853	17.400	0.843	-1.328
RF-7	16.318	0.836	17.186	0.840	18.554	0.856	17.293	0.837	-0.798
ELMP-3	16.683	0.866	17.708	0.873	18.855	0.877	17.725	0.867	-1.570
ELMP-7	16.525	0.856	17.464	0.861	18.856	0.879	17.581	0.860	-0.700

The top two performed models are shaded; The ETS forecasts are used as the baseline for calculating AvgRelMAE

In Fig. 7, the 95% confidence intervals of Nemenyi ranking test for the AvgRelMAE of 17 models in the test data are displayed. GBRT-7 is on average ranked as the best performance model, but the 95% interval of the GBRT-7 overlaps with that of RF-7 and GBRT-3, implies that RF-7 and GBRT-3 are not significantly worse than GBRT-7. But the intervals for the other models are outside the right interval boundary of GBRT-7 without any overlap, and hence, these models perform significantly worse than the GBRT-7.

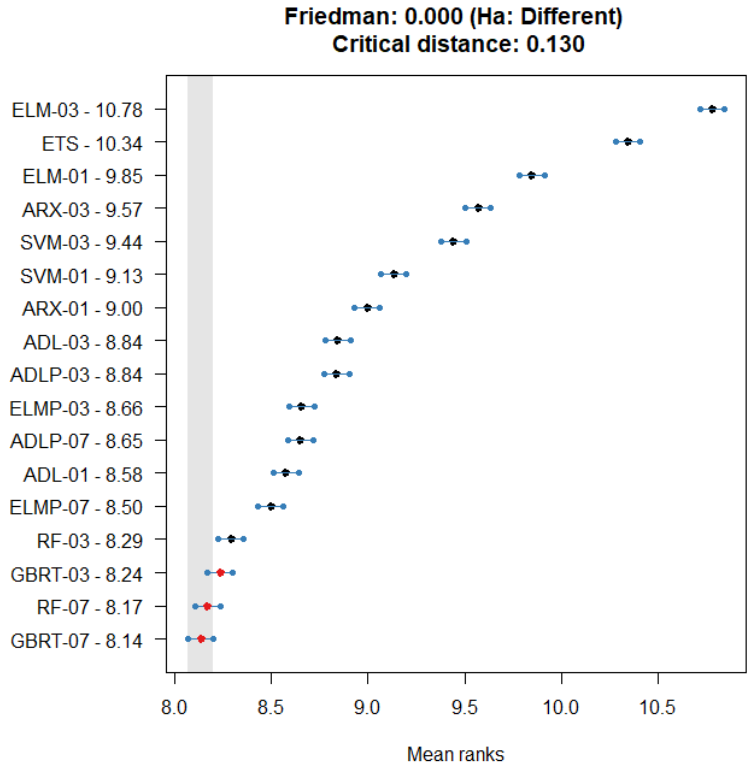

Figure 7. Nemenyi ranking test with AvgRelMAE at 5% significance level on test data

In order to train the meta-learners, we selected 9 models among the 17 models listed in Table 6 and 7 to build a pool of base-forecasters, based on their performance on the training data (Table 6). This consists of five individual forecasting models: ETS, ADL-1, ARX-1, ELM-1, SVM-1, and four pooled forecasting models, including ADLP-3, RF-7, GBRT-7 and ELM-7. Figure 8 shows that the proportion of the sales time series for which a base-forecaster performs as the best (evaluated with sMAPE).

We see that the proportions where a forecaster performed best compared to all others are in general close in both the training and test periods, and the distribution of the proportions among base-forecasters is nearly balanced. It is interesting to find that ETS, though performing the worst among all the base-forecasters on average (Table 6 & 7), outperformed its competitor forecasters most often.

Figure 8 Proportions of the sales time series for which a particular base-forecaster performs as the best

5.2 Forecasting performance of the meta learners

The forecasts generated by the selected base-forecasters are further processed by nine meta-learners and three simple combination methods. The results are reported in Table 8, and this time the best performing base-forecaster, i.e. GBRT-7, works as the baseline for calculating AvgRelMAE. The meta-learner M3 uses all nine individual forecasting models and M4 uses all eight pooled forecasting models listed in Table 6 as the base-forecasters respectively.

The meta-learner M0 provides the most accurate forecasts measured by all the accuracy metrics over all the horizons; the low bias measurement is also outstanding among all the meta-learners. Meta-learner M1 which uses hand-selected features, together with M2 which uses automatic feature learning but without extracting information from influential factors, also provide superior forecasts than the other meta-learners, simple combination methods, and GBRT-7. While M5 which targets the individual selection of the best performing base-forecaster cannot even beat the aggregate performance of some base-forecasters, all the meta-learners integrated with ensemble methods provide better forecasts than the two simple ensemble benchmarks, as well as GBRT-7. This shows the importance of ensemble methods being included in the meta-learner. FFORMA2 performs better than FFORMA1 which shows the small additional value of the features extracted from the influential factors, but the performance is worse than M1 though both of them use the same set of hand-selected features.

Another important finding is that meta-learners using the pool of base-forecasters either composing of only individual forecasting models (M3) or of pooled forecasting models (M4) perform poorly compared to those meta-learners using mixed models (M0 & M1). M4, especially, shows very limited improvements over GBRT-7, though GBRT-7 is one of its base-forecasters. The results show the

importance of using a pool of mixed base-forecasters in meta-learning when forecasting SKU level sales.

Among the three ensemble benchmarks, the first two combiners, i.e., E1 and E2, perform better than the baseline, and close to some of the meta learners, e.g., FFORMA1, M4 and M6. But the combiner E3, the combination of four best performing forecasters in the training set, including GBRT-7, RF-7, GBRT-7 and RF-7 which are all pooling methods, has very similar performance to the baseline. The comparisons of forecasting performance between E1 and E3 indicate that even traditional combination methods could benefit potentially from the combining of a mixed pool of base-forecasters (i.e. including forecasts from both pooling and individual base-forecasters).

Table 8 Forecasting performance of nine meta-learners and three ensemble benchmarks in the test data

Meta-learner	Horizon								
	h=1		h=4		h=7		h=1-7		
	sMAPE	AvgRelMAE	sMAPE	AvgRelMAE	sMAPE	AvgRelMAE	sMAPE	AvgRelMAE	MPE
M0	15.953	0.987	16.747	0.980	17.965	0.960	16.849	0.968	-0.170
M1	15.980	0.989	16.765	0.981	17.972	0.964	16.865	0.970	-0.046
M2	15.980	0.989	16.771	0.983	17.981	0.963	16.870	0.970	-0.034
M3	16.183	0.997	17.114	1.001	18.514	0.993	17.231	0.995	-1.117
M4	16.140	0.999	16.950	0.992	18.125	0.971	17.053	0.982	0.394
M5	17.067	1.058	18.073	1.069	19.203	1.035	18.135	1.050	3.886
M6	16.128	1.002	16.939	0.994	18.050	0.965	17.006	0.979	-0.077
E1	16.171	1.004	16.985	0.997	18.161	0.977	17.078	0.986	-0.064
E2	16.103	1.001	16.900	0.990	18.079	0.968	16.994	0.978	-0.292
E3	16.237	1.001	17.169	1.006	18.501	0.992	17.247	0.996	-1.334
FFORMA1	16.060	0.992	16.868	0.985	18.110	0.969	16.975	0.977	-0.060
FFORMA2	16.023	0.992	16.824	0.983	18.049	0.965	16.928	0.974	0.254

The best performance models are shaded; The GBRT-7 forecasts are used as the baseline for calculating AvgRelMAE

Fig. 9 shows the Nemenyi test intervals calculated based on the ranks of AvgRelMAE with all methods in Table 8. It shows that though M0 has lowest mean rank, it is not significantly better than M1 and M2. But it is obvious that the meta-learner M0, M1 and M2 show significantly better performance than all the others. Perhaps surprisingly, M2 shows strong performance despite not using the influential features as shown in Table 5.

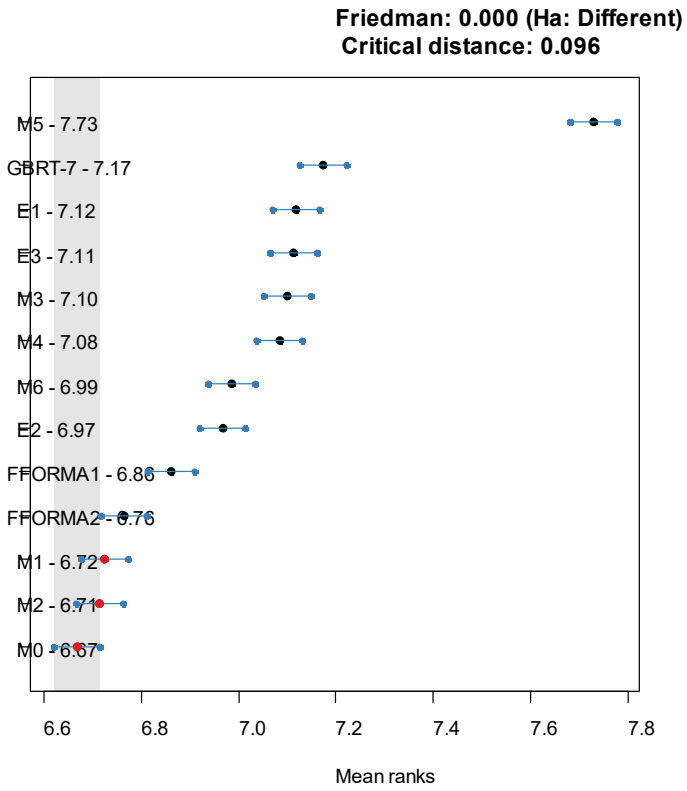

Figure 9 Nemenyi test at 5% significance level on nine meta-learners, three simple combination methods and GBRT-7

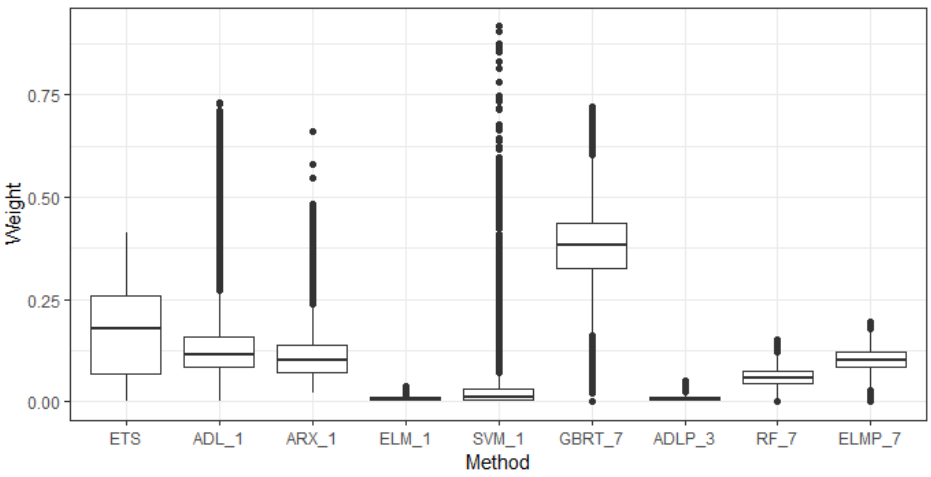

Figure 10. The boxplot of the weights of nine base-forecasters used by M0 when forecasting test periods.

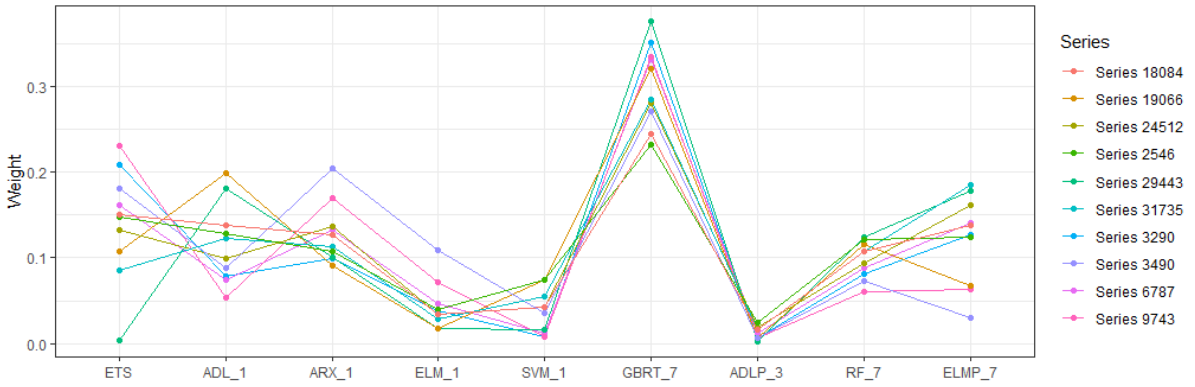

Figure 11. The parallel coordinates plot on the weights of nine base-forecasters in M0 for forecasting ten randomly selected sale series

To obtain a deeper understanding on the performance of the M0, in Figure 10, we depict the boxplots of the weights of nine base-forecasters on forecasting test periods. It shows that GBRT-7 is on average given largest weight averaged over time. Three individual forecasting methods including ETS, ALD-1 and ARX-1 also achieve more than 10 percent weight on average. ELM-1 and ADLP-3 are among the lowest contributors. To give a better insight into how the nine base forecasts are combined for different sales series, Figure 11 depicts the combination weights in M0 for forecasting a sample of sales time series. Both figures provide further evidences that, when using meta-learning to forecast SKU sales (or even in more general settings when forecasting many time series. e.g. Makridakis et al., 2019), it is better to pool base-forecasters composing of simple individual forecasting methods (with fewer lags) and complex pooled forecasting methods (with more lags).

Table 10 reports the forecasting performance of the various meta learners segmented into promotion and non-promotion weeks separately. The promotion here is defined as meeting at least one of three conditions: price lower than the median of the prices during training periods, existing display, or feature advertising. All the methods' relative forecasting performances in the two segments are in general consistent with their results of the full-sample evaluation reported in Table 9. M0 still outperforms all the others across all accuracy measures in both segments which shows its robust forecasting performance. We find that all the methods have better relative forecasting performance in promotion weeks than in non-promotion weeks. In promotion weeks, except for M3 and M5, all the meta learners deliver an additional three to six percent of improvements over the baseline, compared to their performance in non-promotional weeks as measured by AvgRelMAE.

To explore the interpretability of the learned features, in this paper's supplementary material, we provide some visualization results on how the convolutional filters work in M0. In the first convolutional layer, some filters are explainable easily, but after three layers, the features extracted are too complex to be interpreted. From their correlation table, we could not find any obvious corresponding relations between hand-selected features and automatically extracted features.

Table 10 Forecasting performance of eight meta-learners and three ensemble benchmarks in promotion and non-promotion periods

	Promotion		Non-promotion	
	AvgRelMAE (to ETS)	AvgRelMAE (to GBRT-7)	AvgRelMAE (to ETS)	AvgRelMAE (to GBRT-7)
M0	0.760	0.950	0.830	0.984
M1	0.762	0.952	0.831	0.985
M2	0.762	0.951	0.832	0.986
M3	0.794	0.992	0.842	0.998
M4	0.770	0.962	0.843	0.999
M5	0.836	1.044	0.890	1.055
M6	0.772	0.964	0.838	0.993
E1	0.780	0.974	0.845	1.001
E2	0.770	0.962	0.838	0.993
E3	0.795	0.994	0.842	0.999
FFORMA1	0.771	0.963	0.834	0.989
FFORMA2	0.770	0.962	0.833	0.987

The best performance model is shaded.

Similarly, Table 11 reports the forecasting performance of the various meta learners for existing and new SKUs. New SKUs here refer to the SKUs that are sold in a store in the test periods but are not sold in the same store in the training periods. Similar to the results reported in the Table 10, M0 again showed robust forecasting performance, it outperforms all the others across all accuracy measures in both SKU segments. All the methods have better forecasting performance for existing SKUs than their respective performance for new SKUs. Most of the meta learners obtain larger improvements over the baseline when forecasting existing SKUs compared to their performance for forecasting new SKUs in terms of AvgRelMAE.

Table 11 Forecasting performance of eight meta-learners and three ensemble benchmarks for existing and new SKUs

	Existing SKUs		New SKUs	
	AvgRelMAE (to ETS)	AvgRelMAE (to GBRT-7)	AvgRelMAE (to ETS)	AvgRelMAE (to GBRT-7)
M0	0.814	0.966	0.843	0.975
M1	0.816	0.968	0.845	0.977
M2	0.816	0.968	0.845	0.977
M3	0.839	0.995	0.861	0.995
M4	0.826	0.980	0.856	0.990
M5	0.886	1.051	0.904	1.045
M6	0.824	0.978	0.853	0.986
E1	0.829	0.984	0.859	0.992
E2	0.823	0.976	0.853	0.986
E3	0.839	0.996	0.861	0.995
FFORMA1	0.822	0.975	0.851	0.983
FFORMA2	0.820	0.973	0.848	0.980

New SKUs here refer to the SKUs that are sold in a store in the test periods but are not sold in the same store in the training periods.

Table 12 Forecasting performance of the eight meta-learners and three ensemble benchmarks over six categories (evaluated with AvgRelMAE, the GBRT-7 forecasts are used as the baseline)

	Milk	Beer	Mayo	Coffee	Yogurt	Laundet
M0	0.947	0.986	0.970	0.975	0.967	0.964
M1	0.949	0.986	0.973	0.976	0.969	0.967
M2	0.947	0.988	0.972	0.976	0.969	0.967
M3	0.994	0.998	0.985	0.999	0.995	0.997
M4	0.954	0.995	0.983	0.990	0.986	0.980
M5	1.017	1.062	1.051	1.034	1.063	1.038
M6	0.956	0.991	0.979	0.984	0.983	0.978
E1	0.958	0.996	0.986	1.007	0.987	1.003
E2	0.959	0.991	0.978	0.988	0.977	0.979
E3	0.995	0.997	0.987	1.001	0.996	1.000
FFORMA1	0.957	0.989	0.977	0.981	0.979	0.973
FFORMA2	0.953	0.988	0.970	0.977	0.977	0.971

In Table 12, we compare the forecasting results of the meta learners and benchmark methods for different categories, evaluated with AvgRelMAE and used the GBRT-7 as the baseline. In general, M0 consistently outperform all the benchmark methods across all six categories. But the extent of the improvements varies among different categories. In the category Milk, it could improve the forecasts over the baseline more than 5 percent on average, while in category Beer, it can only achieve relatively limited forecasting improvements.

5.3. Discussion

We empirically showed the advantages of the proposed meta learner with respect to several variants and state-of-the-art approaches for store SKU weekly sales forecasting tasks. Now we discuss the results to answer the research questions listed in Section 4.

(AQ1) The proposed meta-learner (M0) has significant superior forecasting performance over all the base forecasters and the simple average combinations of the forecasters. On average, M0 has 3.2 percent improvements over the best performing base-forecaster (measured by AvgRelMAE in Table 9). The meta-learner is particularly effective during promotion weeks, which has 5 percent improvements over the best performing base-forecaster, compared to 1.6 percent of improvements in non-promotion weeks (Table 10). This could be explained by the observation that sales in non-promotional weeks are relatively stable and therefore easier to be forecasted. The meta-learner could improve the forecasting accuracy for both existing and new SKUs significantly, but is relatively more effective for existing SKUs. It is particularly effective for fast moving categories, and less effective for slow moving categories.

(AQ2) Compared to FFORMA, the proposed meta-learner (M0) showed significantly superior forecasting performance (Figure 9). This indicates that the proposed method can provide better

combined forecasts than FFORMA.

(AQ3) We find that the meta-learner using supervised feature learning (M0) consistently performs better than the meta-learner using unsupervised hand-selected features (M1), though the Nemenyi test did not show significant superiority. But another benefit of the automatic feature learning is that we no longer need to worry about the problem of how to extract features from time series under study before using meta-learning methods, so this simplifies the data processing effort and makes the meta-learning approach easy to implement.

(AQ4) We find that the meta-learner which learns features from both the time series of sales and influential factors can improve forecasting performance potentially over the meta-learner using only sales time series as the input, though the ranking based Nemenyi test did not show the improvements to be significant (or impactful). One explanation of this results is that most of the variations due to the influential factors have already been reflected in the sales series, so influential factors contain limited additional information for the meta-learner to select the best ensembles.

Meta leaners including FFORMA and M6, and the combiner E2, are all combining base forecasters according to their past accuracy statistics. They have similar forecasting performance and are all superior to the best base forecaster, but they are all inferior to the proposed meta learner and its variants.

(AQ5) We find that the meta-learners using a mixed pool of base-forecasters can improve the forecasting performance significantly over meta-learners using base-forecasters consisting of only individual or alternatively pooled forecasting methods. This is consistent with Smyl (2020), the winner of M4-competition, that a hybrid model using the two modeling strategies can improve forecasting performance.

(AQ6) Though it is common and straightforward to target selecting the best performing base-forecaster in meta-learner designing (Table 1), our findings show that this is a misleading practice. Our results highly recommend choosing to target the best combination of base-forecasters when using meta-learning for forecasting retail sales.

6. Conclusions

Product level sales forecasting in retail is essential to sound retail business planning to improve their service performance in daily operations. Taking advantage of the huge amount of historical data accumulated by retailers, this paper is the first to evaluate the performance of meta-leaning methods in forecasting SKU×Store weekly sales. We proposed a novel meta-learner based on convolution neural networks, which can extract features automatically from raw sales time series and their influential factors using a supervised learning approach. This research is also the first to propose using a mixed pool of base-forecasters which includes forecasting methods of using both individual forecasting and pooled forecasting strategies. In addition to the novel meta-learning methodology we described, we

obtained a series of empirical findings through our forecasting experiments, which are important for guiding retail forecasting practice.

In general, the proposed meta-learner can improve retail sales forecasting accuracy significantly. When design a meta learner to forecast retail sales, we recommend (1) the use of base-forecasters including both individual and pooled forecasting methods; (2) targeting finding a best combination forecasts instead of identifying just the best one; (3) considering the use of supervised feature learning instead of handcraft features; and (4) considering the extraction of features from external influential factors in addition to sales time series. We note that the differences between just including the sales history and the both time series of sales and the influential factors proved in this application to be small, suggesting that for longer horizons or in some applications greater benefit from considering exogenous features might prove more valuable.

While the differences between some of the individual meta-learners are small compared to the base forecasters, the gains in retail applications should prove valuable, translating directly into better service/ lower inventory. They also suggest a route forward in software improvements where the methods employed for retail data are often based on simple selection routines and, for example, do not include the pooled methods which have proved so beneficial. Implementation and user acceptance may prove problematic as expert judgmental adjustment remains a common practice in retail forecasting (Fildes, et al., 2020). Whether such machine learning methods as those described here can deliver better value than those currently practiced remains an important research issue.

The results, as with any empirical study, suffer from the limitations of using a particular data source. In retailing, additional retail data including different categories, SKUs and, of course, retailers including on-line, would help generalize the findings we report. Further research could also explore more sophisticated automatic feature extraction methods for time series data or apply our approach to many other practical forecasting scenarios, e.g., energy or financial forecasting applications, areas also concerned with the problems of forecasting many related time series with external influential factors. It would also be an interesting empirical question whether our conclusions would hold when the proposed meta learners are tested on more general univariate time series data like that found in the M3 or M4 competitions.

Acknowledgments The authors acknowledge the help from the anonymous referees which has led to further clarification of the results. The first author acknowledges the support of the National Natural Science Foundation of China under grant no. 71571089.

References

Aburto, L., & Weber, R. (2007). Improved supply chain management based on hybrid demand forecasts. *Applied*

Soft Computing, 7(1), 136-144. doi:10.1016/j.asoc.2005.06.001

Ainscough, T. L., & Aronson, J. E. (1999). An empirical investigation and comparison of neural networks and regression for scanner data analysis. *Journal of Retailing and Consumer Services, 6*(4), 205-217. doi:http://dx.doi.org/10.1016/S0969-6989(98)00007-1

Allen, P. G., & Fildes, R. (2001). Econometric forecasting. In J. S. Armstrong (Ed.), *Principles of Forecasting* (pp. 303-362): Norwell, Ma: Kluwer.

Andrews, R. L., Currim, I. S., Leeflang, P. S. H., & Lim, J. (2008). Estimating the SCAN∗PRO model of store sales: HB, FM or just OLS? *International Journal of Research in Marketing, 25*(1), 22-33. doi:http://dx.doi.org/10.1016/j.ijresmar.2007.10.001

Arunraj, N. S., & Ahrens, D. (2015). A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics, 170*, 321-335.

Barrow, D. K., & Kourentzes, N. (2016). Distributions of forecasting errors of forecast combinations: Implications for inventory management. *International Journal of Production Economics, 177*, 24-33. doi:https://doi.org/10.1016/j.ijpe.2016.03.017

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798-1828. doi:10.1109/TPAMI.2013.50

Blattberg, R. C., & George, E. I. (1991). Shrinkage Estimation of Price and Promotional Elasticities: Seemingly Unrelated Equations. *Journal of the American Statistical Association, 86*(414), 304-315. doi:10.2307/2290562

Boylan, J. E., Goodwin, P., Mohammadipour, M., & Syntetos, A. A. (2015). Reproducibility in forecasting research. *International Journal of Forecasting, 31*(1), 79-90. doi:10.1016/j.ijforecast.2014.05.008

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32. doi:10.1023/a:1010933404324

Bronnenberg, B. J., Kruger, M. W., & Mela, C. F. (2008). Database Paper: The IRI Marketing Data Set. *Marketing Science, 27*(4), 745-748.

Cerqueira, V., Torgo, L., Pinto, F., & Soares, C. (2017). *Arbitrated Ensemble for Time Series Forecasting.* Paper presented at the (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2017. Lecture Notes in Computer Science, Cham.

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system*. Paper presented at the Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining.

Dekker, M., van Donselaar, K., & Ouwehand, P. (2004). How to use aggregation and combined forecasting to improve seasonal demand forecasts. *International Journal of Production Economics, 90*(2), 151-167. doi:http://dx.doi.org/10.1016/j.ijpe.2004.02.004

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1-30.

Fildes, R., Ma, S., & Kolassa, S. (2020). Retail forecasting: research and practice. *International journal of forecasting, in press*.

Fildes, R., & Petropoulos, F. (2015). Simple versus complex selection rules for forecasting many time series. *Journal of Business Research, 68*(8), 1692-1701. doi:https://doi.org/10.1016/j.jbusres.2015.03.028

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189-1232. doi:10.1214/aos/1013203451

Friedman, J. H., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software, 33*(1), 1-22. doi:10.18637/jss.v033.i01

Fumi, A., Pepe, A., Scarabotti, L., & Schiraldi, M. M. (2013). Fourier Analysis for Demand Forecasting in a Fashion Company. *International Journal of Engineering Business Management, 5*, 1-10. doi:10.5772/56839

Gür Ali, Ö., Sayin, S., van Woensel, T., & Fransoo, J. (2009). SKU demand forecasting in the presence of promotions. *Expert Systems with Applications, 36*(10), 12340-12348. doi:10.1016/j.eswa.2009.04.052

Gür Ali, Ö., & Yaman, K. (2013). Selecting rows and columns for training support vector regression models with large retail datasets. *European Journal of Operational Research, 226*(3), 471-480. doi:10.1016/j.ejor.2012.11.013

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.* Paper presented at the 2015 IEEE International Conference on Computer Vision (ICCV).

Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2019). Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1-1. doi:10.1109/TPAMI.2019.2913372

Huang, T., Fildes, R., & Soopramanien, D. (2014). The value of competitive information in forecasting FMCG retail

product sales and the variable selection problem. *European Journal of Operational Research, 237*(2), 738-748. doi:http://dx.doi.org/10.1016/j.ejor.2014.02.022

Hyndman, R., Kang, Y., Montero-Manso, P., Talagala, T., Wang, E., Yang, Y., . . . Moorman, J. R. (2019). tsfeatures: Time Series Feature Extraction. R package R package v.1.0.1.

Hyndman, R., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. *2008, 27*(3), 22. doi:10.18637/jss.v027.i03

Kück, M., Crone, S. F., & Freitag, M. (2016). *Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data.* Paper presented at the 2016 International Joint Conference on Neural Networks (IJCNN).

Kalaoglu, O. I., Akyuz, E. S., Ecemis, S., Eryuruk, S. H., Sumen, H., & Kalaoglu, F. (2015). Retail demand forecasting in clothing industry. *Tekstil Ve Konfeksiyon, 25*(2), 174-180.

Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A. v. d., Graves, A., & Kavukcuoglu, K. (2016). Neural Machine Translation in Linear Time. *arXiv:1610.10099*.

Kalousis, A., & Theoharis, T. (1999). NOEMON: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intelligent Data Analysis, 3*(5), 319-337. doi:https://doi.org/10.1016/S1088-467X(99)00026-8

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *arXiv:1408.5882*.

Kingma, D. P., & Ba, J. (2015). *Adam: A Method for Stochastic Optimization*. Paper presented at the 3rd International Conference for Learning Representations, San Diego.

Koning, A. J., Franses, P. H., Hibon, M., & Stekler, H. O. (2005). The M3 competition: Statistical tests of the results. *International journal of forecasting, 21*(3), 397-409. doi:https://doi.org/10.1016/j.ijforecast.2004.10.003

Kourentzes, N. (2019). tsutils: Time Series Exploration, Modelling and Forecasting. R package v. 0.9.1.

Lang, S., Steiner, W. J., Weber, A., & Wechselberger, P. (2015). Accommodating heterogeneity and nonlinearity in price effects for predicting brand sales and profits. *European Journal of Operational Research, 246*(1), 232-241. doi:10.1016/j.ejor.2015.02.047

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time-series *The handbook of brain theory and neural networks*. Arbib, M.A.: MIT Press.

LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). *Convolutional networks and applications in vision.* Paper presented at the Proceedings of 2010 IEEE International Symposium on Circuits and Systems.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). *Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations*. Paper presented at the Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Quebec, Canada.

Lemke, C., & Gabrys, B. (2010). Meta-learning for time series forecasting and forecast combination. *Neurocomputing, 73*(10), 2006-2016. doi:https://doi.org/10.1016/j.neucom.2009.09.020

Levy, M., Grewal, D., Kopalle, P., & Hess, J. (2004). Emerging trends in retail pricing practice: implications for research. *Journal of Retailing, 80*(3), xiii-xxi. doi:10.1016/j.jretai.2004.08.003

Lu, C.-J. (2014). Sales forecasting of computer products based on variable selection scheme and support vector regression. *Neurocomputing, 128*, 491-499. doi:http://dx.doi.org/10.1016/j.neucom.2013.08.012

Ma, S. H., & Fildes, R. (2020). Forecasting third-party mobile payments with implications for customer flow prediction *International journal of forecasting, In press*.

Ma, S. H., Fildes, R., & Huang, T. (2016). Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra- and inter-category promotional information. *European Journal of Operational Research, 249*(1), 245-257. doi:http://dx.doi.org/10.1016/j.ejor.2015.08.029

Makridakis, S., & Petropoulos, F. (2019). The M4 competition: Conclusions. *International Journal of Forecasting*. doi:https://doi.org/10.1016/j.ijforecast.2019.05.006

Makridakis, S., & Petropoulos, F. (2020). The M4 competition: Conclusions. *International journal of forecasting, 36*(1), 224-227. doi:https://doi.org/10.1016/j.ijforecast.2019.05.006

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., & Lin, C.-C. (2019). e1071: Misc Functions of the Department of Statistics. R package v. 1.7-2.

Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., & Talagala, T. S. (2018). *FFORMA: Feature-based Forecast Model Averaging*. Working Paper 19/18.

Mouselimis, L., & Gosso, A. (2018). elmNNRcpp: The Extreme Learning Machine Algorithm. R package v 1.0.1.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., . . . Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. *arXiv:1609.03499*.

Pillo, G. D., Latorre, V., Lucidi, S., & Procacci, E. (2016). An application of support vector machines to sales

forecasting under promotions. *4or Quarterly Journal of the Belgian French & Italian Operations Research Societies, 14*(3), 309-325.

Prudêncio, R. B. C., & Ludermir, T. B. (2004). Meta-learning approaches to selecting time series models. *Neurocomputing, 61*(1), 121-137.

Ramos, P., & Fildes, R. (2017). *Characterizing retail demand with promotional effects*. Paper presented at the International Symposium on Forecasting Cairns, Australia.

Ramos, P., Santos, N., & Rebelo, R. (2015). Performance of state space and ARIMA models for consumer retail sales forecasting. *Robotics and Computer-Integrated Manufacturing, 34*, 151-163. doi:10.1016/j.rcim.2014.12.015

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2019). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*. doi:https://doi.org/10.1016/j.ijforecast.2019.07.001

Santos, c. N. D., & Zadrozny, B. (2014). *Learning character-level representations for part-of-speech tagging*. Paper presented at the Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, Beijing, China.

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International journal of forecasting, 36*(1), 75-85. doi:https://doi.org/10.1016/j.ijforecast.2019.03.017

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research, 15*(6), 1929-1958.

Talagala, T., Hyndman, R., & Athanasopoulos, G. (2018). *Meta-learning how to forecast time series*. Retrieved from https://EconPapers.repec.org/RePEc:msh:ebswps:2018-6

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks, 10*(5), 988-999. doi:10.1109/72.788640

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., & Gomez, A. N. (2017). *Attention is all you need*. Paper presented at the NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, United States.

Wang, X., Smith-Miles, K., & Hyndman, R. (2009). Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series. *Neurocomputing, 72*(10), 2581-2594. doi:https://doi.org/10.1016/j.neucom.2008.10.017

Widodo, A., & Budi, I. (2013). *Model selection using dimensionality reduction of time series characteristics*. Paper presented at the International Symposium on Forecasting, Seoul, South Korea.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67-82. doi:10.1109/4235.585893

Wong, W. K., & Guo, Z. X. (2010). A hybrid intelligent model for medium-term sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm. *International Journal of Production Economics, 128*(2), 614-624. doi:10.1016/j.ijpe.2010.07.008

Wright, M. N., & Ziegler, A. (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *Journal of Statistical Software, 77*(1), 17. doi:10.18637/jss.v077.i01

Xia, M., Zhang, Y., Weng, L., & Ye, X. (2012). Fashion retailing forecasting based on extreme learning machine with adaptive metrics of inputs. *Knowledge-Based Systems, 36*, 253-259. doi:10.1016/j.knosys.2012.07.002

Yu, Y., Choi, T.-M., & Hui, C.-L. (2011). An intelligent fast sales forecasting model for fashion products. *Expert Systems with Applications, 38*(6), 7373-7379. doi:10.1016/j.eswa.2010.12.089

Zębik, M., Korytkowski, M., Angryk, R., & Scherer, R. (2017). *Convolutional Neural Networks for Time Series Classification.* Paper presented at the International Conference on Artificial Intelligence and Soft Computing Cham.

Zheng, Y., Liu, Q., Chen, E., Ge, Y., & Zhao, J. L. (2014). *Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks.* Paper presented at the Web-Age Information Management, Cham.

Ziegler, A., & König, I. R. (2014). Mining data with random forests: current options for real-world applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4*(1), 55-63. doi:10.1002/widm.1114

Zotteri, G., & Kalchschmidt, M. (2007). A model for selecting the appropriate level of aggregation in forecasting processes. *International Journal of Production Economics, 108*(1–2), 74-83. doi:http://dx.doi.org/10.1016/j.ijpe.2006.12.030