2020

On Cross-Series Machine Learning Models

Xiaodan Zhu
William & Mary - Arts & Sciences, zxdan523@gmail.com

On Cross-Series Machine Learning Models

Xiaodan Zhu

Shijiazhuang, Hebei, China

Bachelor of Science, Jiangnan University, China, 2012
Master of Engineer, University of Electronic Science and Technology of China, China, 2015

A Dissertation presented to the Graduate Faculty
of The College of William & Mary in Candidacy for the Degree of
Doctor of Philosophy

Department of Computer Science

College of William & Mary
August 2020

APPROVAL PAGE

This Dissertation is submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

Xiaodan Zhu

Xiaodan Zhu

Approved by the Committee, April 2020

Zhenming Liu

Committee Chair
Zhenming Liu, Assistant Professor, Computer Science
College of William & Mary

Weizhen Mao

Weizhen Mao, Professor, Computer Science
College of William & Mary

Pieter Peers, Associate Professor, Computer Science
College of William & Mary

Bin Ren

Bin Ren, Assistant Professor, Computer Science
College of William & Mary

Anh Ninh, Assistant Professor, Mathematics
College of William & Mary

Yanhua Li, Assistant Professor, Computer Science
Worcester Polytechnic Institute

ABSTRACT

Sparse high dimensional time series are common in industry, such as in supply chain demand and retail sales. Accurate and reliable forecasting of high dimensional time series is essential for supply chain planning and business management. In practical applications, sparse high dimensional time series prediction faces three challenges: (1) simple models cannot capture complex patterns, (2) insufficient data prevents us from pursuing more advanced models, and (3) time series in the same dataset may have widely different properties. These challenges prevent the currently prevalent models and theoretically successful advanced models (e.g., neural networks) from working in actual use.

We focus our research on a pharmaceutical (pharma) demand forecasting problem. To overcome the challenges faced by sparse high dimensional time series, we develop a cross-series learning framework that trains a machine learning model on multiple related time series and uses cross-series information to improve forecasting accuracy. Cross-series learning is further optimized by dividing the global time series into subgroups based on three grouping schemes to balance the tradeoff between sample size and sample quality. Moreover, downstream inventory is introduced as an additional feature to support demand forecasting. Combining the cross-series learning framework with advanced machine learning models, we significantly improve the accuracy of pharma demand predictions.

To verify the generalizability of cross-series learning, a generic forecasting framework containing the operations required for cross-series learning is developed and applied to retail sales forecasting. We further confirm the benefits of cross-series learning for advanced models, especially RNN. In addition to the grouping schemes based on product characteristics, we also explore two grouping schemes based on time series clustering, which do not require domain knowledge and can be applied to other fields. Using a retail sales dataset, our cross-series machine learning models are still superior to the baseline models.

This dissertation develops a collection of cross-series learning techniques optimized for sparse high dimensional time series that can be applied to pharma manufacturers, retailers, and possibly other industries. Extensive experiments are carried out on real datasets to provide empirical value and insights for relevant theoretical studies. In practice, our work guides the actual use of cross-series learning.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Dr. Zhenming Liu, for his guidance and support during these years. Special thanks go to the members of my defense committee, Professor Weizhen Mao, Professor Pieter Peers, Professor Bin Ren, Professor Anh Ninh and Professor Yanhua Li, for their support and help during my defense.

Second, thanks to Professor Anh Ninh and Professor Hui Zhao for their effort and instruction on my research project. This dissertation would not have been possible without their help. I also thank Professor Andreas Stathopoulos, Professor Pieter Peers, and Dr. Ivan Medvedev for their guidance on my other research projects. They gave me a lot of inspiration and advice that will affect my life.

Third, I would like to thank our Computer Science administration team, including Vanessa Godwin, Jacqulyn Johnson, and Dale Hayes. They helped me solve all aspects of study and life in my Ph.D. career.

Lastly, thanks to my lab members, Qiong Wu, Zheng Zhang and Sirui Wang. Specially, thanks to my friends, Qingsen Wang, Zhaoliang Duan, Sirui Wang, Jianing Zhao, Xing Gao, Steven Goldenberg, and Zeyi Tao. Thanks for their company.

I would like to dedicate this dissertation to my parents Mr. Jialin Zhu and Ms. Liu Liu for their endless love, support and company, and my grandpa Liu and grandpa Zhu for their blessing.

LIST OF TABLES

LIST OF FIGURES

On Cross-Series Machine Learning Models

Chapter 1

Introduction

This dissertation develops a collection of machine learning techniques optimized for sparse high-dimensional time series. Sparse high-dimensional time series are settings in which a model needs to produce forecasts for multiple time series (i.e., a vector time series) and in which the data is not sufficiently large, such that standard models (e.g., Vector Autoregression, Random Forest, etc.) may fail to work [7, 70, 34]. Despite the practical importance of sparse high-dimensional time series and their potential impacts, there has been a limited amount of research to tackle this problem.

High-dimensional time series are common in many fields, including economics, finance, functional genomics, neuroscience, and climatology. Applications include stock market return inferences [34, 65, 14], gene regulatory network reconstruction [82], identification of connections in different brain areas [106], the study of atmospheric processes [80], and more. These applications require a large number of temporally observed variables based on a relatively small sample size (i.e., the number of time points). In addition to sample size limitations, another type of time series contains a large number of zeros, making them more difficult to predict. This type of time series often appears in supply chain demand [119, 22], retail sales [6, 96], energy consumption [4], etc. Therefore, models commonly used in other fields cannot provide accurate predictions for these applications.

In industry, using machine learning models to generate accurate and reliable predictions for high-dimensional time series is critical [34, 22, 6]. Manufacturers need demand forecasts for supply chain planning, power plants need energy consumption forecasts for resource allocation, retailers need sales forecasts for business management, among others. However, the currently used methods in industry cannot meet users' requirements [116]. Due to the lack of proper learning approaches for sparse high-dimensional time series, advanced machine learning models have not exploited their potential advantages. Therefore, it is essential to investigate the challenges in sparse high-dimensional time series forecasting problems and find generalizable solutions.

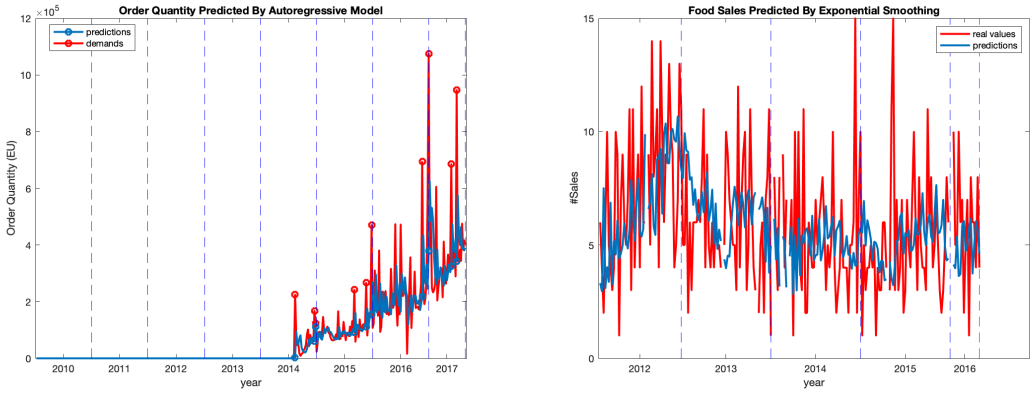
1.1 Problem Statement

Through investigations in the industry and preliminary experiments on real datasets, we find that sparse high-dimensional time series prediction mainly faces three problems in practical applications: (1) *simple models fail to capture complex patterns*, (2) *insufficient data prevents us from pursuing more advanced models*, and (3) *time series in the same dataset may have widely different properties*. These problems prevent the current prevalent methods in the industry from providing reliable predictions, and theoretically successful advanced models fail to work in actual use.

- *Simple models fail to capture complex patterns*.

  Recently, Weller and Crone surveyed 200 companies and found that univariate methods have maintained their dominant position in the industry [116]. In particular, exponential smoothing (EST), moving average (MA), and autoregressive (AR) models are the most popular machine learning models. However, the simple patterns captured by these models are not adequate to describe complex temporal behaviors. For example, Figure 1.1(a) shows the order quantities of a drug (red) and the predictions (blue) of the autoregressive model. Due to latent factors such as the market environment and distri-

bution strategies, the order quantity of some drugs may have an extremely high peak at a certain time point. We call this phenomenon ”spikes”, which is defined quantitatively in Chapter 3. As shown in the figure, simple models like AR have difficulty capturing the spikes indicated by the red circles. In addition to the ”spikes”, under the influence of periodic market activities and special events, time series may also have long-term and short-term seasonal patterns [4]. The traditional way to model time series seasonality is to use statistical forecasting models such as EST and autoregressive integrated moving average (ARIMA) [49, 12]. Figure 1.1(b) compares the sales of a food (red) with the EST forecast (blue). Although EST can recognize different types of seasonality (e.g., additive and multiplicative), the actual situations are more complicated.



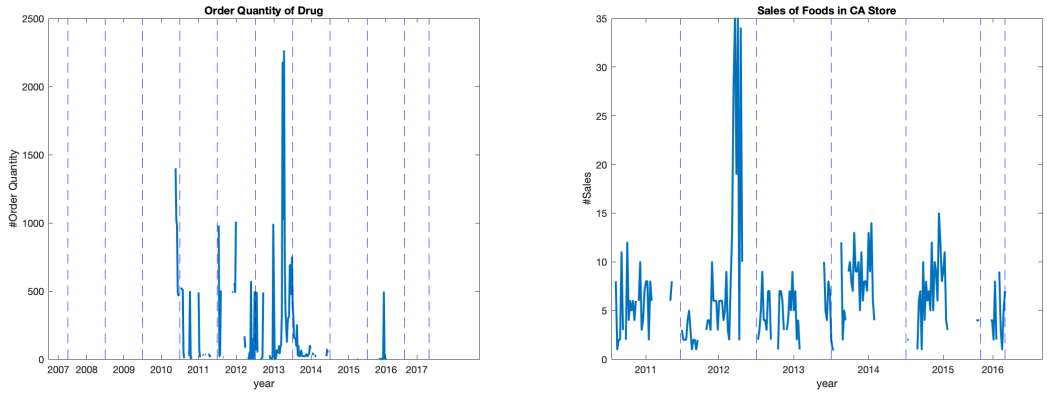
(a) Drug Demand Predicted By AR (b) Food Sales Predicted By EST

Figure 1.1: Complex patterns

Another high-dimensional time series forecasting method is to use structural models like vector autoregression (VAR) [105, 59]. Unlike using univariate forecasting models in which the parameters are estimated independently for each time series, VAR learns the contemporaneous relationships between different temporal variables [12]. However, VAR fails to capture non-linear patterns, and the cross-correlations between different time series are weak in some scenarios [22]. Moreover, the sparse high-dimensional time series introduces too many insignificant parameters without sufficient training data. As a result, VAR suffers from a severe overfitting problem.

- *Insufficient data prevents us from pursuing more advanced models*.

  To capture complex and non-linear patterns, some applications use more advanced machine learning models, such as support vector regression (SVR) [75, 84, 97], random forest (RF) [68, 33], and artificial neural network (ANN) [71, 40, 63, 121]. However, according to the results of the Makridakis Competitions (M competitions) [78], the actual performance of many sophisticated models (e.g., ANN) is not as good as that of the simple models (e.g., EST and AR), mainly because training and testing sophisticated machine learning models require a large amount of data for better estimation of the model parameters in order to achieve a better performance [32, 6]. Yet, this condition is not readily met in most forecasting contexts, since there exists a temporal constraint, that is, old data from the distant past may have little value to the current prediction task (this is in fact confirmed in our data analysis). In addition, some products may have been on the market for just a short time, leaving many zeros in the records. Figure 1.2 shows the examples of the drug order quantity and the food sales in our datasets. Building machine learning models based on these time series will cause overfitting problems.



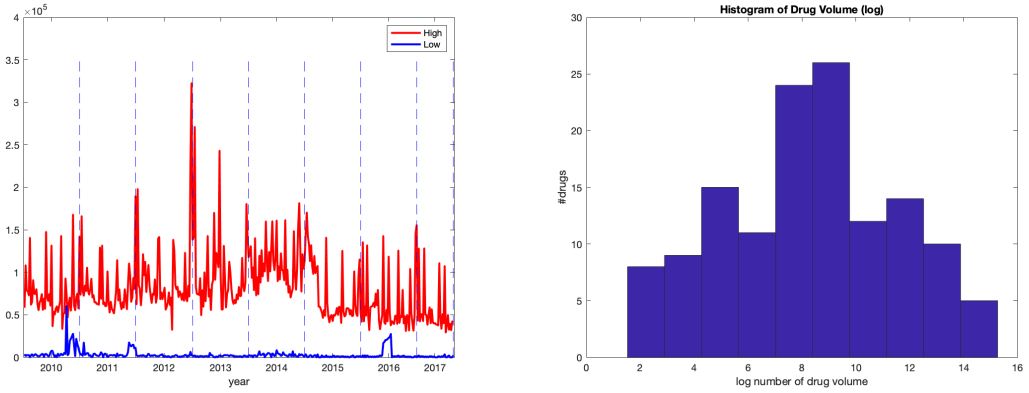
(a) Drug Order Quantity (b) Food Sales

Figure 1.2: Large Fraction of Zeros

- *Time series in the same dataset may have widely different properties*.

  To overcome the insufficient data problem, some studies [22, 6] have proposed similar

ideas as ours, that is, to jointly train a machine learning model using the time series of different products. However, in some cases, the properties of different product time series vary greatly. The model trained on global time series may perform poorly on certain individual time series. For example, a model that performs well for high volume (i.e., average of order quantity) drugs may fail to work on low volume drugs, partly because the training process tends to optimize the performance of the samples (e.g., high volume drugs) that have the most impact on the loss function. Figure 1.3(a) shows examples of high volume and low volume drugs time series. The histogram of the drug volume (in log scale) in one of our datasets is shown in Figure 1.3(b).



(a) High volume & Low volume

(b) Histogram of (log) drug volume

Figure 1.3: Time series with different properties

1.2 Overview

We motivate our research on sparse high-dimensional time series prediction by presenting a real business case in pharmaceutical demand forecasting. Then, we provide a detailed description of datasets used in the thesis including two datasets on pharma demands and one dataset on retail sales. For clarity of presentation, we shall start the analysis using the first pharma dataset, which shows significant benefits of learning from multiple relevant time-series in improving forecasting accuracy. With the remaining datasets,

we demonstrate that the proposed approach can be conveniently generalized as a powerful framework applicable to a broader range of forecasting situations. Our methodology stems from the idea of training a machine learning model on related time series and using cross-series information to improve prediction accuracy. We call it "*cross-series learning*". Combining optimized cross-series learning strategies with advanced machine learning models, we achieve significant improvements compared to the benchmarks.

- *Cross-series learning*.

  Nowadays, with the help of advanced data collection and storage technology, most companies have large amounts of business data resources, for example the sales of hundreds of products in retail, the energy consumption of thousands of households, the load for servers in a data center, and more. The rapid increase in data quantity does not mean that the individual time series has effective information available, however. First, as mentioned, for time-varying patterns, only the most recent data is useful. Second, the series may contain long runs of constant values due to the mismatch between the frequency of data variation and the sampling rate. Third, records of newly launched and non-consumable products contain a large fraction of zeros. However, there are a large number of similar time series available. The limitation of not being able to expand vertically to the past motivates us to look horizontally across related time series. Learning from related time series not only allows us to obtain cross-series information, but it also provides sufficient data for more advanced models. In actual use, since we built one model for multiple time series, cross-series learning can also save time and labor costs for model selection and hyperparameters tuning.

- *Grouping schemes*.

  Cross-series learning is based on the assumption that the related time series share the same temporal patterns. However, in real case, even similar products may have different behaviors, such as the demand for drugs used for different diseases, the sales of seasonal vegetables in different seasons, the electricity consumption of users in differ-

ent regions, etc. In addition, as described in Section 1.1, training models on global time series can affect the performance of specific individual time series. Hence, we optimize cross-series learning by building separate models on groups of similar time series and use different criteria to measure the similarity of the time series. For example, we use the Anatomical Therapeutic Chemical (ATC) code to group drugs according to the domain knowledge of the pharma industry. Without domain knowledge, we can also use the distance between time series (e.g., the Euler distance and dynamic time warping (DTW)) to construct clusters. By using grouping schemes, we can balance the tradeoff between the sample size and sample quality for each model.

- *Additional features*.

  In addition to the cross-series information from related time series, we can introduce additional features with strong contemporaneous relationships to help enhance forecasting accuracy. For example, downstream inventory is related to demand [18, 123], and price fluctuations may affect product sales [124, 104]. We can also generate new features using the time series itself, such as exponential moving average [65], trend and seasonal components [4, 5], etc. According to different applications, we collect and generate various types of features and conduct extensive experiments to verify their effectiveness.

1.3 Contributions

The main contribution of this dissertation is three-fold.

First, this dissertation proposes a cross-series learning framework for sparse high-dimensional forecasting which addresses three challenges: (a) Tradeoff between sample size and sample quality for advanced models: We propose cross-product training to resolve the lack of data issue and various grouping schemes to guarantee sample quality based on different rationales including domain knowledge; (b) Additional features: For

pharma demand forecasting, we introduce two key features, downstream inventory levels and considerations of supply chain structure information, inspired by the operations literature, and design how to effectively include these features; (c) Model efficiency and interpretability: Nonlinear auto-regressive/recurrent models (RNN) significantly outperforms other options (e.g., VAR, tree-based machine learning models). Using domain knowledge and numerical analysis, we also provide possible explanations of the effectiveness of the best performing model (RNN).

Second, using two pharma demand datasets with hundreds of drugs and one retail sales dataset with hundreds of food products, we validate the superior performance of our proposed model framework. More importantly, our work provides important empirical value and insights. For example, we test the value of downstream inventory information and supply chain structure information, which has been discussed in theoretical operations literature [18, 123], but not empirically tested.

Third, our cross-series forecasting model framework (including grouping schemes, using additional features such as downstream inventory and supply chain structure information, in combination with the RNN models) can be applied to other manufacturers, wholesalers, and possibly other industry based on its robust performances. Domain knowledge is important for making modifications to this framework when adapting to other industries.

The dissertation also provides practical guidelines of executing such a framework in reality. Interactions with industry leaders (such as Google) have confirmed the value of this work. Indeed, machine learning, as a new data-driven method, has the potential of capturing (external and internal) hidden factors affecting economics time series forecast. To some extent, it will replace/enhance some of the human experts' functions, providing more accurate and consistent forecasts. Given that machine learning has made its way into other areas in industry, e.g., product development, marketing, it is encouraging to see that companies are also open to machine learning methods for demand and sales forecasting (Section 2.5). Based on the results from our questionnaire, the top pharma companies we interacted with indicated that they are open to machine learning forecast-

ing models (compared to traditional models like linear regression) as long as there is a significant accuracy improvement ($> 10\%$). Our results show that RNN has performances superior to that of linear regression and the discussion of the possible reasons for the effectiveness of RNN also adds its interpretability. We hope that this provides important information for pharma companies to make informed decisions/tradeoffs.

1.4 Dissertation Organization

The remainder of the dissertation is organized as follows. In Chapter 2, we detail the machine learning models for high-dimensional time series forecasting, review the literature, position our paper among the related works and show the current state of demand forecasting in the pharma industry. In Chapter 3, we present the research setting along with a detailed description of our datasets and identify new features (e.g., inventory information). In Chapter 4, We use advanced machine learning models and domain knowledge to develop an optimized cross-series learning framework for drug demand forecasting. Chapter 5 extends the pharma demand forecasting framework to a generic version, which is verified on another pharma demand dataset and retail sales dataset. We conclude the dissertation in Chapter 6 with a summary of important insights. Detailed results that cannot be presented in the dissertation body due to space limitations are provided in the Appendix.

Chapter 2

Background

First, we provide some preliminary definitions and concepts in section 2.1 to facilitate our discussion in subsequent chapters. Then we give a brief review of commonly used machine learning models for time series prediction in section 2.2. Some of these models will later serve as the benchmark for our proposed forecasting method. In section 2.3, we discuss potential problems when applying these machine learning models to our application settings and intuitively outline a solution approach to mitigate their impacts. Closely related works in recent years are summarized in section 2.4, which also sets the background for our proposed research agenda in the next chapter. Finally, in section 2.5, we present the current state of demand forecasting in pharma industry based on our surveys and interactions with the top 5 pharma manufactures.

2.1 Preliminaries

A time series is a sequence of observations, each recorded at time t [16]. There are different types of time series, according to the time parameter t. In our research, we mainly focus on discrete time series with equally spaced time intervals. In other words, for continuous observations on a time series, the time point of an observation is a monotonically increasing sequence with the same step size. Equation (2.1) shows an example with $T+1$

time points.

$$\begin{aligned} &\{t_0, t_1, \cdots, t_i, \cdots, t_j, \cdots, t_T\} \\ &\forall i, j \in [0, T), t_i < t_{i+1}, t_{i+1} - t_i = t_{j+1} - t_j \end{aligned} \tag{2.1}$$

For convenience, we set the time interval to 1 by default. Therefore, the above time series can also be expressed as $\{t_0, t_0 + 1, t_0 + 2, \cdots, t_0 + T\}$.

If the time series has only one time-dependent variable, we call it a **univariate time series** and denote the observation at time t as x_t. Equation (2.2) shows an example of a univariate time series recorded from t to $t + T$.

$$\{x_t, x_{t+1}, \cdots, x_{t+T}\} \tag{2.2}$$

If there is more than one variable, e.g., x_t and y_t, and the variables are interrelated, we call it a **multivariate time series** and denote the observation at time t as (x_t, y_t) [12]. Equation (2.3) shows an example of a multivariate time series recorded from t to $t + T$.

$$\{(x_t, y_t), (x_{t+1}, y_{t+1}), \cdots, (x_{t+T}, y_{t+T})\} \tag{2.3}$$

We can also treat the multivariate time series as a high-dimensional univariate time series. Each observation is a high-dimensional vector, such as $\boldsymbol{x}_t = (x_t, y_t)$. Therefore, we use $\{\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \cdots, \boldsymbol{x}_{t+T}\}$ to express a general time series including both univariate and multivariate time series.

An intrinsic feature of time series is that adjacent observations are dependent. Based on the nature of this dependence, **Time series prediction** uses p available observations before time t from a time series to forecast its value at some future time $t + h$. Equation

(2.4) shows a general form of time series prediction model.

$$\boldsymbol{x}_{t+h} = f(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \cdots, \boldsymbol{x}_{t-p+1}) \tag{2.4}$$

where p is the order of the model and h is called the horizon or leading time [12].

2.2 Machine Learning Models for Time Series Prediction

Since time series forecasting plays an important role in many application areas such as economics, supply chain management, neuroscience, genomics, etc. This research topic has been highly active with much development in both theory and methods. For example, Exponential Smoothing, the Autoregressive Model, Support Vector Regression, and the Boost Regression Tree, are well-established and widely-used forecasting methods, which can often provide robust and accurate prediction in many situations. In particular, in the last few decades, neural networks have achieved remarkable results across fields; and thus, many deep learning models have also been developed for time series prediction, such as Long-Short Term Memory, Gated Recurrent Units, and Temporal Convolutional Nets. In what follows, we present the most commonly used series prediction models in detail.

2.2.1 Exponential Smoothing

Exponential Smoothing (EST) is one of the most widely used forecasting models [42], proposed by Brown, Holt, and Winters in the 1950s [17, 49, 117]. The basic assumption is that an observation in a time series is a weighted sum of the preceding observations and the weight decays exponentially as the observations get older. EST models are widely used in business and industry to extrapolate different types of patterns in the univariate time series. Two main patterns captured by EST are trend and seasonality. A **Trend**

is when the time series as a whole has an upward or downward trend over time, while **Seasonality** is when the time series shows a periodic change.

Equation (2.5) shows the simple exponential smoothing model that includes the forecast equation and level equation.

$$\begin{aligned} \text{Forecast Equation:} & x_{t+h} = l_t \\ \text{Level Equation:} & l_t = \alpha x_t + (1-\alpha) l_{t-1} \end{aligned} \tag{2.5}$$

where $0 \leq \alpha \leq 1$ is the smoothing parameter. When we recursively substitute older times for the level equation, we get the expanded expression in Equation (2.6).

$$x_{t+h} = \alpha x_t + \alpha(1-\alpha) x_{t-1} + \alpha(1-\alpha)^2 x_{t-1} + \cdots \tag{2.6}$$

From Equation (2.6), we can see that the weights of past observations decay with the rate of $1-\alpha$.

Based on the simple exponential smoothing model, Holt and Winters also add a trend component and seasonality components to capture corresponding patterns [49]. Pegels further categorized each trend and seasonality into additive and multiplicative types [90]. Based on Pegels' classification, Gardner added another damped type to trend [42]. Hyndman et al. provided a categorization of the 15 EST models depending on the types of patterns recognized by each model (e.g., Holt-Winters additive model and Holt-Winters multiplicative model) [54]. By combining different components, we can get a variety of exponential smoothing models. In practice, we will select the model with the best performance.

2.2.2 Moving Average

In a moving average model, future observations are constructed from weighted sums of past forecast errors [45]. Equation (2.7) shows the pth-order moving average process.

$$x_{t+h} = b + \omega_0 \epsilon_t + \omega_1 \epsilon_{t-1} + \cdots + \omega_{p-1} \epsilon_{t-p+1} \tag{2.7}$$

where b is the expected mean from past observations, ω_i is the weight of the ith forecast error and ϵ_{t-i} is white noise.

When making predictions, the simple moving average is the unweighted past values that can smooth out the short-term fluctuations and capture the long-term trend of the time series. Equation (2.8) is a simple moving average with a window size of p.

$$\hat{x}_{t+h} = \frac{x_t + x_{t-1} + \cdots + x_{t-p+1}}{p} \tag{2.8}$$

Since more recent observations often have more impact, higher weights are assigned to the recent observations and a weighted average is used to make predictions. Like exponential smoothing, Equation (2.9) shows the construction of an exponential moving average with an order of p.

$$\hat{x}_{t+1} = \alpha x_t + (1 - \alpha)\hat{x}_{t-1} \tag{2.9}$$

where $\alpha = \frac{2}{p+1}$ is the decay rate. When we recursively substitute the smoothing term $\hat{x}_{t-i}$, Equation (2.9) can also be written in the form of Equation (2.10).

$$\hat{x}_{t+1} = \alpha x_t + \alpha(1-\alpha)\hat{x}_{t-1} + \cdots + (1-\alpha)^{p-1} x_{t-p+1} \tag{2.10}$$

In this form, we can see the exponential moving average is a special case of exponential smoothing [117].

2.2.3 Autoregressive Models

In addition to EST and MA, autoregressive (AR) models are also the best-known time series forecasting models. Slutsky, Walker and Yaglom [86] first established the concept of AR based on Yule's idea of modeling time series using a stochastic process [120].

Autoregressive Models aim to describe the autocorrelations between observations [45]. The basic autoregressive model predicts the variable of interest by using the linear combination of past values of the variable itself. This is why it is called **autoregression**. Equation (2.11) shows the basic autoregressive model with an order of p.

$$x_{t+h} = b + \omega_0 x_t + \omega_1 x_{t-1} + \cdots + \omega_{p-1} x_{t-p+1} + \epsilon_t \tag{2.11}$$

where b is the expected mean of observations and ϵ_t is white noise. An autoregressive model is always used to predict stationary time series whose properties do not vary with time. In other words, for any $T > 0$, the distribution of $(x_t, x_{t+1}, \cdots, x_{t+T})$ does not depend on t [12]. To process non-stationary time series, there is a model integrating autoregressive models with moving average models, called the autoregressive integrated moving average (ARIMA) [13].

One way to predict multivariate time series is to extend the model of univariate time series. Take the Autoregressive Model as an example. Suppose we have a bivariate time series, as described in Equation (2.3). x is the variable of interest we want to predict. We

can use both x and y as predictor variables to build a pth-order autoregressive model, which is shown as Equation (2.12).

$$\begin{aligned} x_{t+h} &= b + \theta_0 x_t + \phi_0 y_t + \theta_1 x_{t-1} + \phi_1 y_{t-1} + \\ &\quad \cdots + \theta_{p-1} x_{t-p+1} + \phi_{p-1} y_{t-p+1} + \epsilon_t \\ &= b + \begin{bmatrix} \theta_0 \\ \phi_0 \end{bmatrix}^T \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \begin{bmatrix} \theta_1 \\ \phi_1 \end{bmatrix}^T \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + \begin{bmatrix} \theta_{p-1} \\ \phi_{p-1} \end{bmatrix}^T \begin{bmatrix} x_{t-p+1} \\ y_{t-p+1} \end{bmatrix} + \epsilon_t \\ &= b + \boldsymbol{\omega}_0^T \cdot \boldsymbol{x}_t + \boldsymbol{\omega}_1^T \cdot \boldsymbol{x}_{t-1} + \cdots + \boldsymbol{\omega}_{p-1}^T \cdot \boldsymbol{x}_{t-p+1} + \epsilon_t \end{aligned} \tag{2.12}$$

where $\boldsymbol{\omega}_i = [\theta_i, \phi_i]^T$ and $\boldsymbol{x}_i = [x_i, y_i]^T$.

The disadvantage of the model described in Equation (2.12) is that it only considers the relationship between one variable of interest and the remaining variables, but not the interrelationships between all variables. In the 1980s, Sims introduced a more generalized multivariate AR model called Vector Autoregression (VAR) to model the contemporaneous relationships between variables [105]. Suppose we have N variables $x_{1,t}$, $x_{2,t}$, $\cdots$, $x_{N,t}$, Equation (2.13) is a pth-order VAR model.

$$\begin{bmatrix} x_{1,t+h} \\ x_{2,t+h} \\ \vdots \\ x_{N,t+h} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} + \boldsymbol{W}_0 \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ \vdots \\ x_{N,t} \end{bmatrix} + \boldsymbol{W}_1 \begin{bmatrix} x_{1,t-1} \\ x_{2,t-1} \\ \vdots \\ x_{N,t-1} \end{bmatrix} + \cdots + \boldsymbol{W}_{p-1} \begin{bmatrix} x_{1,t-p+1} \\ x_{2,t-p+1} \\ \vdots \\ x_{N,t-p+1} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

$$\Updownarrow$$

$$\boldsymbol{x}_{t+h} = \boldsymbol{b} + \boldsymbol{W}_0 \boldsymbol{x}_t + \boldsymbol{W}_1 \boldsymbol{x}_{t-1} + \cdots + \boldsymbol{W}_{p-1} \boldsymbol{x}_{t-p+1} + \boldsymbol{\epsilon} \tag{2.13}$$

where each W_τ is a $N \times N$ parameter matrix for vector $\boldsymbol{x}_{t-\tau}$. VARs have been widely used for economic time series analyses.

In addition to the models commonly used in the industry (e.g., EST, MA and VAR), many other machine learning models have also achieved significant results in time series

prediction. We mainly study three of them: tree-based models, kernel-based models and artificial neural network.

2.2.4 Tree-based Models

Ensemble learning is a technique that combines basic learners to produce a powerful model [46]. One of the commonly used basic learners is the decision tree. Many decision tree-based regression models are generated by using different ensemble methods. Bagging and boosting are two well-known ensemble methods. Bagging (i.e., bootstrap aggregation) creates multiple learners by using the new training sets randomly sampled from the original set and taking the average of the predictions from the learners as the final result [15]. Random Forest (RF) is a representative bagging model that is widely used for time series forecasting. Boosting is similar to bagging, but the learners learn from the residual generated by the last iteration [37]. Equation (2.14) shows an example of this boost procedure.

$$
\begin{aligned}
f^{(i+1)}(x_t, x_{t-1}, \cdots, x_{t-p+1}) &= f^{(i)}(x_t, x_{t-1}, \cdots, x_{t-p+1}) + h^{(i)}(x_t, x_{t-1}, \cdots, x_{t-p+1}) \\
h^{(i)}(x_t, x_{t-1}, \cdots, x_{t-p+1}) &= x_{t+h} - f^{(i)}(x_t, x_{t-1}, \cdots, x_{t-p+1})
\end{aligned}
\tag{2.14}
$$

where $f^{(i)}(.)$ is the model at the ith iteration, and $h^{(i)}$ is the predictive residual at the ith iteration.

Tree-based models make predictions by referring to the results of different learners so that they can provide robust predictions and prevent overfitting. Even applied to simple prediction models like AR or EST, it can still effectively improve predictive accuracy [46].

2.2.5 Kernel-Based Models

Kernel-based learning methods are a class of pattern analysis algorithm that map instances to a high-dimensional space by using different kernel functions. Instead of learning fixed parameters for the input features, kernel learning models study the similarity of instances in the implicit feature spaces [83]. Therefore, we do not need to calculate data coordinates in the high-dimensional space, but simply evaluate the dot product of each pair of instances. This approach is called the **kernel trick**.

Many kernel-based regression models, such as Support Vector Regression (SVR) [108] and the Gaussian Process (GP) [107], can also be used for time series prediction. The easiest way to do so is to treat the segments of time series as observations and build a kernel learning model with general kernel functions, such as the Radial Basis Function (RBF) and polynomial kernel.

SVR, for example, learns a high-dimensional linear model by using the data mapped from a p-dimensional space, as shown in Equation (2.15).

$$x_{t+h} = \boldsymbol{\omega}^T \phi([x_t, x_{t-1}, \cdots, x_{t-p+1}]^T) + b \tag{2.15}$$

where $\boldsymbol{\omega}^T$ and b are the parameters of the model and $\phi(.)$ is the function mapping the p-dimensional data to a high-dimensional space. Suppose there is a kernel function $k(.,.)$, defined as Equation (2.16).

$$\begin{aligned} &\forall \boldsymbol{x}_1 = [x_{1,t}, x_{1,t-1}, \cdots, x_{1,t-p+1}]^T, \boldsymbol{x}_2 = [x_{2,t}, x_{2,t-1}, \cdots, x_{2,t-p+1}]^T \\ &k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \phi(\boldsymbol{x}_1)^T \cdot \phi(\boldsymbol{x}_2) \end{aligned} \tag{2.16}$$

Then, the loss function of SVR is optimized in the dual space and Equation (2.17) is used to make predictions.

$$\hat{x}_{t+h} = \sum_{i=1}^{N} \lambda_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{2.17}$$

where λ_i is the parameter in dual space, N is the number of training instances, $\boldsymbol{x}$ is the time series segment to predict, and $\boldsymbol{x}_i$ is the segment from the ith training time series. Kernel Ridge Regression (KRR) [115] is another regression model, with an identical form to Equation (2.15) but a different loss function.

2.2.6 Artificial Neural Networks

At present, artificial neural networks (ANN) have achieved remarkable results in many fields, including time series prediction. A multilayer perceptron (MLP) is a feedforward ANN in which the inputs are filtered through multiple hidden layers. The activation functions between the hidden layers introduce non-linearity to the output, which enables ANN to capture complex patterns [94]. Compared with MLP, recurrent neural networks (RNN) are more suitable for learning time series patterns due to their unique feedback architecture [38]. The key to time series prediction is finding the time dependency between observations. Therefore, the information needs to persist during the learning process. The chain-like nature of an RNN can address this issue, as shown in Figure 2.1.

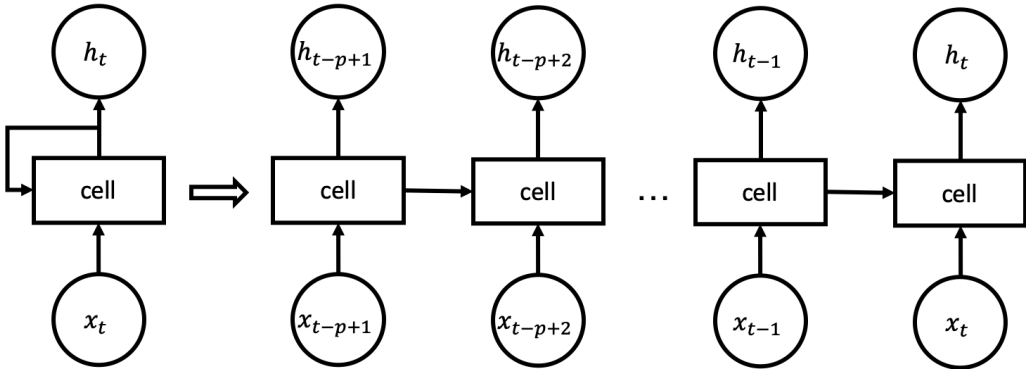


Figure 2.1: Structure of recurrent neural network

The structure on the left-hand side is a basic RNN cell with a feedback connection; the unrolled structure is on the right-hand side. h_t is the hidden state generated at the tth step and passed to the next successor. By using this structure, the RNN can memorize the information from historical observations [99].

However, the basic RNN model has exploding and vanishing gradient problems, especially when dealing with ”long-term” dependencies. Hochreiter and Schmidhuber introduced a new RNN architecture called Long Short-Term Memory (LSTM) [48], which is composed of an input gate, output gate, and forget gate, as shown in Figure 2.2.

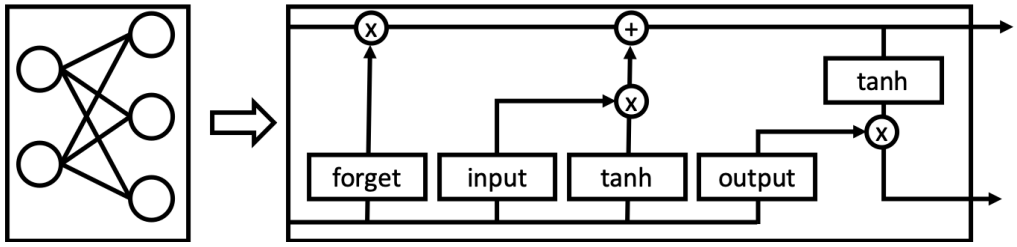


Figure 2.2: Long short-term memory cell

The standard RNN cell is constructed from simple neural network layers, as shown on the left-hand side. The structure of LSTM is shown on the right-hand side. It uses three gates, which are sigmoid functions, to control the information flow. By using gates and simple elementwise operations, it can decide which information is retained, which is forgotten, and which is outputted. Therefore, LSTM can avoid long dependency problems.

Based on LSTM, Kyunghyun Cho proposed a new cell architecture called the Gated Recurrent Unit (GRU) [27]. GRU has fewer parameters because it combines the forget gate and input gate. On smaller data sets, GRU performs better than LSTM, and it is becoming more popular due to its simpler architecture. Beyond RNN-based models, many researchers also use Convolutional Neural Networks (CNN) to process time series data. A Temporal Convolutional Network (TCN) uses a hierarchy of temporal convolutional filters to capture long-range patterns [110]. Aaron van den Oord et al. proposed WaveNet, which is constructed from a stack of causal convolutional layers and uses a dilated convo-

lution filter to save computational costs [87]. LSTM and CNN (LSTM-CNN) are also used together, so that the CNN learns time series structural information and LSTM detects temporal dependency [103].

For many standard time series datasets, the previously described machine learning models can give accurate predictions. But in practice, many data sets are not so ideal. Common problems include the following: 1. There is insufficient data, 2. The data contains too many missing values, 3. The data is unevenly distributed. 4. The data contains patterns that change over time, such that data from the distant past is less useful for predicting the most recent patterns. In these cases, machine learning models built using individual time series may not give reliable predictions. Especially for complex deep learning models, there may not be enough data to fit their parameters, and in many cases, this will lead to overfitting problems. In section 2.3, we will summarize the methods to solve such problems developed in recent years by using cross-series information, list their advantages, and point out their deficiencies or inadequacies.

2.3 Potential Pitfalls

Machine learning models, especially deep learning models, rely heavily on data. If the amount of data is insufficient or the data is not properly processed, many theoretically successful models will perform poorly in practical applications [122]. This is especially true in time series prediction. Due to the limited data from an individual time series, the prediction results of many complex models are even worse than those of simple models [78]. Even if we have a long time series, it does not mean that we have sufficient data. For example, in a time series, data can be dominated by certain specific patterns, making other patterns difficult to capture during the learning process. Another common situation is that data from the distant past is of little help in learning recent patterns, while short-term data is not sufficient for the task.

In the face of this problem, the most direct solution is to increase the amount of data.

One approach is to use multivariate time series, such as VAR models and many deep learning models [113, 21, 58, 24]. However, the increase in the amount of data will simultaneously increase the complexity of the model. In the end, the amount of data may still be insufficient. Moreover, additional variables may not be helpful in predicting the target variable. Another approach is to use related or similar time series. In many cases, related individuals often follow similar patterns. By sharing data across different time series of related individuals, not only the problem of insufficient data can be effectively solved, but the impact of outliers can also be reduced, thereby making more robust models.

Trapero et al. propose a forecasting model which pools past information from other stock-keeping units when there is not enough promotional history available for the current one [114]. This model directly uses historical information from other time series. There will be problems when applying it to other datasets, since arbitrarily fitting all time series into one model may lose the focus of the current time series patterns. The reason for this is that during the training process, machine learning models tend to satisfy the data that has the greatest impact on the loss. Therefore, if time series with similar patterns to the current one do not dominate or the training data contains too many unrelated patterns, it will make the learning process extremely difficult. Therefore, before fitting the time series into the model, we need to group them and let time series belonging to the same group share a model. This process requires a trade-off between the size of the group and the similarity of the time series within the group. We want to learn as many patterns as possible while ensuring sufficient training data.

Chapados develops a hierarchy model based on a Bayesian framework and lets the time series in one subgroup share the same model parameters, which achieves good results in supply chain planning [22]. Bandara et al. group related time series to train a new LSTM for sales forecasting. They use two grouping strategies. The first is to use domain knowledge, such as sales rankings and the percentage of zero sales. The other is to use time series clustering. They use k-means to cluster time series according to the feature vectors constructed from handcrafted features, such as trend, spikiness, and lin-

earity [6]. In [5], Bandara et al. investigate more clustering methods, including k-means, DBSCAN, Partition Around Medoids (PAM), and Snob. Salinas et al. introduce a probability forecasting framework called DeepAR, which uses RNN-based architecture to learn from groups of similar time series and provides estimations of the forecast distributions [96].

The methodologies of the works listed above are closely related to ours. However, their clustering metrics are mainly based on handcrafted features, such as mean, trend, and seasonality. In many cases, these features do not guarantee that time series with similar patterns can be extracted. Moreover, some features are not applicable to all types of time series, for example, physical trajectories. Therefore, rather than using handcrafted features, we recommend grouping according to the nature of the time series itself. For a general time series, if it has synchronous sampling and equal length, we can use Euclidean distance. For asynchronous sampling, we can use dynamic time warping distance (DTW) [95]. In addition, we can use edit distance on real sequence (EDR) [26], edit distance with real penalty (ERP) [25], and longest common sequence (LCSS) [61]. For physical trajectories, we can use symmetric segment-path distance (SSPD) to compare the similarity of path shapes [10]. According to the review in [10], no one trajectory distance can be robust for all types of trajectories. [1] introduces a framework called Autowarp to learn warping distance by using an autoencoder that best fits to the training time series.

2.4 Related Works

In this section, we review demand forecasting literature, particularly the interface between machine learning and forecasting in the operations management literature, with a special focus on the recent research motivated by real industry problems using data-driven approach.

Due to its crucial role in production and inventory control [57], demand forecasting

has been extensively studied in the past decades [117, 17, 98, 100, 69, 11]. Practical considerations such as collaborative forecasting partnerships between retailers and manufacturers [2], performance of hierarchical forecasting at different levels of aggregation in the supply chain [67], and combining forecasts from multiple models [44] have also been studied. Most of this research focuses on traditional time-series methodology.

Recent years have seen a great development of machine learning applications across many disciplines due to their remarkable abilities to capture hidden patterns. In forecasting domain, Hill et al. successfully applied neural network models to time-series and achieved much better performance as compared to that from traditional statistical forecasting methods [47]. Recently, the similar type of research has appeared in the operations management literature. While limited in numbers, there is an upward trend in this data-driven research. For example, Carbonneau et al. studied the effectiveness of both machine learning and traditional forecasting methods on simulated and real sales data [20]. They reported that traditional methods work well on simulated data, but are less competitive against more advanced machine learning models on real data. A more recent paper leveraging the power of machine learning in demand forecasting is Cui et al., which uses both the operational data (sales and marketing data) and the social media information to improve the accuracy of daily sales forecasts [73].

Notice that all demand forecasting models discussed so far predict the future demands for a product using its own data, where there may be problems in model parameters estimation when the amount of data is limited. To address this problem as well as to help find the common hidden factors, we leverage information from other products. This idea of cross-learning from other products has been used in new product forecasting in which future sales are predicted from a set of features such as price, brand, style based on comparable products [35, 3]. However, in such settings, the dataset is limited to only similar products and the data is not time-series as mentioned.

Within the context of time-series literature, demand forecasting from related time series have also been studied ranging from tourism demand forecasting, hotel room de-

mand forecasting, to electric power demand forecasting [89, 109, 41]. Regardless of the application settings, a standard assumption is that these time series are organically related to each other, for example the demand for hotel rooms and the number of internet search terms about hotel information in the area. Vector autoregression (VAR) is a well-established econometric method for learning from related time series when making forecasts [105]. However, VAR is very different from what we propose and is not applicable to our problem due to overfitting. Specifically, VAR allows all variables to interact linearly with their own and each other's current and past values (lags). Therefore, when there are many time-series involved with many lags, as intended in our problem for cross-learning, the number of VAR coefficients to be estimated is very large, leading to severe overfitting and larger forecast errors even with regularization. As confirmed by our numerical results, VAR performance is significantly worse than even those from the baselines (see Section 4.2). This is confirmed in our numerical results. Indeed, Hyndman and Athanasopoulos suggests using VAR only for a small number of time-series which are known to be correlated with each other [52]. In addition, VAR only captures linear relationships. As confirmed by our results, there exists significant non-linear relationships in our data.

In terms of using non-demand information to help demand forecast, many recent papers leverage social media data to enhance the performance of forecasting [74, 73, 11, 72, 102]. For further details, we refer the readers to Choi et al. for a review [28]. In this dissertation, we identify other suitable non-demand features and use them to predict demand across products. In particular, our idea of leveraging inventory and supply chain structure information within the cross-drug training framework is based on existing theoretical operations management literature. Indeed, the benefit of using downstream's inventory data to enhance the upstream's optimal production/inventory decisions has been studied in the analytical models [18, 123], but has not been tested empirically. Further, among the possible drivers for different inventory levels in the system, the impact of distribution network structure is significant [19]. In our dissertation, we empirically explore the value of inventory information and supply chain structure (DC-level data) in demand forecasting

using real data in a cross-drug training setting.

There is limited and also relatively primitive academic literature on demand forecasting for pharma products compared to other industries (e.g., tourism, energy, etc.). Besides the monograph by Cook [30], prior studies on pharma demand forecasting are summarized and compared to our study in Table 2.1. Specifically, the first row specifies which tier's demand is forecasted in the respective research because different data might be available and used for that specific tier. The pharma supply chain has multiple tiers, including manufactures, trade partners/wholesaler (TPs), distribution centers (DCs), and point-of-care. It is well-known that demand forecasts become less accurate moving up the chain. We propose to forecast demand at the manufacturer's tier.

Table 2.1: Comparison of pharmaceutical demand/sale forecasting papers in the literature.

	Anusha et al. (2014)	Candan et al. (2010)	Kim et al. (2015)	Merkuryeva et al. (2019)	Nikolopoulos (2016)	Zedeh et al. (2014)	**Our Model**
Tier	Retailer	Retailer	Retailer	Distributor	Point-of-care	Distributor	Manufacturer
Benchmark	Moving average Exp. Smoothing Holt-Winter	None	AR	Moving average Linear regression	Diffusion models ARIMA Exp. smoothing Linear regression	ARIMA	Exp. state-space models Moving Average Linear Regression
Proposed model	None	ANN	VARX	Symbolic regression	None	Graph-based analysis and ANN	Clustering and RNN
Utilized data	Historical sales	Historical sales	Historical sales Social network	Historical sales Price	Historical sales (prescription data)	Historical sales	Historical demand Downstream inv. Supply chain info.
Forecast frequency	Monthly	Quarterly	Monthly	Weekly	Yearly	Monthly	Weekly
Forecasting horizon	1-month ahead	1-year ahead	1-month ahead	1-week ahead	1-5 year ahead	1-month ahead	1-2 month ahead
Metrics	MAD, MSE MAPE	None	Prediction error rate	R^2 MAD	R^2 ME, MAE, MSE	R^2 MSE, MAE	NME, NMAE, NMSE
Number of NDCs	2	1	4	1	11	217, but only 21 were analyzed	133 (1st dataset) 112 (2nd dataset)
Time-series cross-validation	No	No	No	No	No	No	Yes

Table 2.1 also compares our dissertation to other research works in terms of benchmarks, models proposed, data utilized, metrics used, and forecasting horizon. The main benchmarks used in these papers are moving average, simple exponential smoothing, and regressions, consistent with those reported in the industry practices described in Section 2.5. In terms of data used, there seems to be a lack of distinction between demand and sales in the pharma forecasting literature (with sales being the right-censored demand by available inventory). All papers in the table except ours used historical sales (instead of actual historical demand) to predict demand, and little other information is

used for forecasting. Specifically, besides historical sales, Kim et al. used customers' response collected in blog documents to help improve drug demand forecasts for a retailer [66]. Merkuryeva et al. used discounted prices in a causal forecasting model to forecast demand at a distributor [81]. The rest only used historical sales. In measuring forecasting accuracy, cross-validation is the standard approach and should be used to ensure the generalizability of the forecasting model to new data. The basic cross-validation procedure involves separating the data into training and test sets, where the training data is used to estimate a forecasting model's parameter, and the test set ($\approx 20\%$ of the observations) is used to evaluate its accuracy [76]. However, this reporting standard is often not used in the pharma forecasting literature. Further, in practice, for accuracy evaluation of forecasting methods, it is recommended to use the more sophisticated time-series cross-validation, where there are a series of test sets and the forecasting accuracy is computed by averaging over these test sets [52]. However, none of these papers implemented time-series cross-validation and most of them only used a small number of drugs.

2.5 Current State of Pharma Demand Forecasting

Based on surveys, reports, and literature, this section describes the current situation of pharma demand forecasting in terms of methods and data used.

Forecasting Methods

In 2018, the global market for pharmaceuticals reached $1.2 trillion, up $100 billion from 2017 (IQVIA Institute for Human Data Science) and the U.S. alone holds over 45% of the global pharma market. Due to the high profit margin - the top 10 pharma companies in the U.S. had a median profit margin of 17% (Angell 2004) - there was a low need for supply chain efficiency and the pharma industry didn't pay much attention to demand forecast until more recently [64, 81]. This seemingly explains the dominant position of simple demand forecasting methods used in the industry. Jain, based on a pharma industry survey, listed the most popular forecasting models as exponential smoothing,

moving averages, and regression [56]. More recently, Weller and Crone surveyed 200 companies (14 of which are pharma companies) and confirmed that univariate statistical methods have maintained their dominant position in pharma and other industries [116]. In particular, exponential smoothing, moving average, and naive methods account for 82.1% of all statistical forecasts. This is true even in the era of using software. Analyzing the results from a joint research initiative of IndustryWeek and SAS, Chase summarized that while companies might be using various softwares to help with demand forecasting, moving average, exponential smoothing, and simple regression models are still the most popular forecasting methods used by the softwares [23].

Cook outlined the typical procedure of demand forecast for in-market pharma products as (1) trending historical data, (2) applying the effects of ex-trend events (i.e., external or internal events that may affect demand but not reflected in the historical data), and (3) converting trended data into forecast outputs based on the first two steps [30]. The challenges to the forecasters are to identify these ex-trend events and quantify the effects of these events on the forecast. While these could be done by human experts' judgments, this is atypical because most pharma companies deal with a large number of national drug codes (NDCs), ranging anywhere from hundreds to thousands of marketed products with different therapeutic characteristics. Even for drugs with the same active pharmaceutical ingredient (API), they may have different dosages, delivery methods (tablets v.s. injection, etc.), corresponding to different NDCs. Thus, forecasting is typically done using software (e.g., SAP, Oracle, R, Excel). One other concern of human judgement is its quality, consistency, and dependence on experiences; hence, human judgement is only incorporated for special cases such as new product launch or known competitor entrance/exit. And, even in the cases when human judgement is incorporated, it is incorporated on top of the algorithm-generated forecasts [56]. Hence, accuracy of algorithm-generated demand forecasting is particularly important.

Data Used

Currently, for statistical forecasting methods, historical sales are the most commonly

used data for forecasting [116]. Benchmarking studies reported in Merkuryeva et al. point out that ”although there is plenty of data useful for more accurate demand forecasting, data usage is limited due to various aspects (e.g., different data formats; lack of data integration tools)” [81]. Chase also observes that despite all the improvements with data collection, downstream data has not been utilized for supply chain demand forecasting and planning [23]. The value of downstream data has been overlooked, even after *supply chain visibility* is made available. Since the mid-2000s, to streamline ordering and purchasing processes in the drug supply chain, Electronic Data Interchange (EDI) has been adopted in the pharma industry. As part of the fee-for-service (FFS) arrangements with the manufacturer, wholesalers must provide inventory data to the manufacturer, typically via the EDI interface known by their numerical designations such as 867, 852, 180 among many others. For instance, EDI 852 contains inventory, product stocking, and product movement records from the trade partners' DCs to the manufacturer [118]. However, there has been a ”lack of new models for increasing forecasting intelligence” [81] and ”minimal investment in the analytic skills of demand planners” [23]. Our dissertation is the first to explore the value of some of this data to pharma demand forecasting tasks.

Forecasting Horizon

The results we provide to the manufacturers are forecasts of future drug demand. In our setting, the future demand refers to the total order quantity of each drug for all distribution centers in a certain week in the future. The number of weeks we predict is called the *forecasting horizon*. According to Cook (2016)[30], the forecasting horizon within the pharma industry can range from short-term, medium-term, to long-term forecasts. The long-term forecasts ($>$ 5 years ahead) are used for strategic planning. For example, to launch a new product 3 years from now, a 10-year forecast is often used. The medium-term forecasts ($>$ 1 year) are produced for financial forecasting and budget planning. The short-term forecasts (daily, weekly, monthly) are involved with operations, such as inventory decisions and manufacturing decisions (procurement of raw materials, scheduling, etc.). We primarily focus on forecasting roles in supply chain operations; hence, the

forecasting horizon considered is 1-8 weeks.

Interactions with pharma companies

To further confirm the current state of demand forecasting in the pharma industry, we interacted with five top pharmaceutical companies whose names are hidden for confidentiality based on a focused questionnaire directed at the points of interest in this dissertation. Specifically, we designed a list of questions in the three aforementioned categories: forecasting methods, data used, forecast horizon, and the use of demand forecast, for which the companies provided answers. Appendix 3 includes the list of specific questions. When doubts existed, further interactions were conducted through interviews or additional correspondences.

All pharma companies we interacted with confirmed that they are using software to forecast demands with simple models such as exponential smoothing (2 companies), moving average (2 companies), and linear regression models (3 companies), and some companies use more than one of the three. One company estimated that human judgment is involved in less than 10% of the cases, while the remaining companies use human judgments in $10-30\%$ of the cases. Moreover, human judgments are primarily used for new product launches and planned promotions. Further, all companies verified that only historical demand is used to generate statistical forecasts. In terms of forecasting horizon, it varies from 3 months (3 companies), 1 month (1 company), to 1 week (1 company). Results of demand forecasts are used in a wide variety of activities. All companies utilize demand forecasts for inventory decisions. Besides, it is also used for cash flow and workforce planning (4 companies), for capacity planning (3 companies), production planning, and promotion planning (3 companies) and for setting sales targets (1 company).

Finally, the pharma companies stated they are not currently using machine learning models, but one is in the early stage of investigating machine learning in demand forecasting. All companies mentioned that they are open to machine learning models as long as these models can bring sufficient improvement ($> 10\%$), even if such models are less interpretable than the currently used models.

Chapter 3

Research Setting and Dataset

This section reviews the research setting for this dissertation. In particular, we first detail our pharma demand datasets in the context of the supply chain network in Section 3.1, followed by a detailed description of the retail sales dataset in Section 3.2.

3.1 Pharma Demand Datasets

3.1.1 Pharma Distribution Network

The pharma supply chain is a complex system in which drugs are delivered from manufacturers to patients through the distribution networks. An over-simplified pharma distribution network highlighting what is pertaining to our work is depicted in Figure 3.1. In particular, the flow of pharma products originates from manufacturers to multiple trade partners (TPs), who then distribute these products via their network of distribution centers (DCs), to downstream point-of-cares (POCs), such as clinics, hospitals, or retail pharmacies. Trade partners can be categorized as “traditional wholesalers” and “specialty distributors”. The former typically have large networks, carry a large variety of drugs, and more often distribute to hospitals, retail pharmacies, and homecare providers, while the latter typically have more controlled networks, specialize in specialty drugs and more often distribute to physician offices, clinics, and independent specialty pharmacies. Regardless of its type,

each trade partner has its own network of DCs, through which POCs receive the drugs.

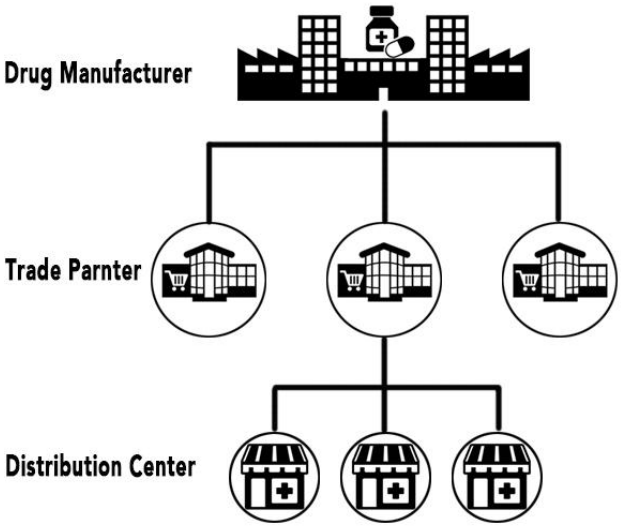


Figure 3.1: Illustration of a pharma distribution network

As mentioned, we obtained from our industry collaborator two large datasets of EDI 852 of two top pharma manufacturers. The datasets consist of supply chain channel data on all pharma products that the respective manufacturer has at the time of data collection. Each product is determined by a unique, three-segment identifier, called NDC. We will focus on analyzing the first dataset and then use the second dataset to confirm our insights in Section 5. We next present more detailed description of the first dataset.

3.1.2 Pharma Demand

Our data includes all transactions over the period from July 2007 to August 2017 between a drug manufacturer and its trade partners' DCs, collected weekly for 133 unique NDCs. Specifically, this refers to information of the quantity sold, the trade partner, the distributor, and the respective inventory level at the corresponding DC. The quantity sold and inventory level from each transaction are measured in pack unit (PU) or extended unit (EU) for each NDC. We choose to use extended unit (i.e., one capsule or tablet for solid

dosage forms or one milliliter for liquid drug products) because this measurement helps to normalize different package sizes, which allows comparisons across NDCs. Specifically, each transaction in the dataset includes information on the quantity sold, the trade partner, the distributor, and the respective inventory level at the corresponding DC. Figure 3.2 shows the format of the data items.

Time(yyyymmdd)	NDC			Trade Partner ID	Distribution Center ID	Order Quantity	Inventory
...							
20120520	ndc1	ndc2	ndc3	3	199157	3.0283e4	1.0611e5
20120527	ndc1	ndc2	ndc3	20	319396	0	180
...							

Figure 3.2: Format of pharma dataset

The first dataset includes 3.4 million transactions, and the second dataset contains 1 million transactions.

Figure 3.3 illustrates a typical time-series of the weekly order quantities of a drug over the period of 2007-2017. The figure shows many demand spikes occurring throughout the years. As will be discussed later, existence of such spikes is due to the prevalent investment buying in the pharma industry. Hence, being able to capture such spikes is important to the performance of forecasting models.

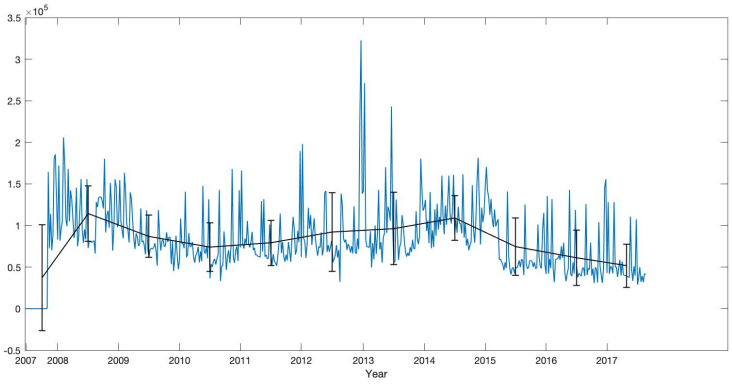


Figure 3.3: Illustration of a drug's order quantities over time

Figure 3.4 shows the sparsity of all drugs' order quantity series. Each row in the figure

indicates whether there were medicines ordered in the week from 2007 to 2017. Blue pixels represent zero order quantities, and green represent non-zero order quantities. In the first dataset, we have 133 unique NDCs with 530 weeks of records, but 52% of the total observations are zeros, which is typical for a sparse high-dimensional time series.

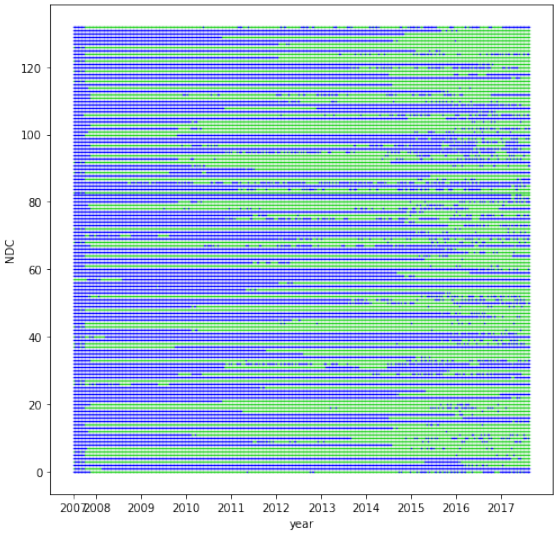


Figure 3.4: Sparsity of the order quantity series

3.1.3 Additional Information

In addition to the past demand information, we use the following non-demand features in our forecasting model.

Inventory Information. The fee-for-service (FFS) contracts prevalent in pharma industry require downstream wholesalers to share inventory information with the manufacturer via EDI 852 interface. Hence, we collect both the historical order quantity and the inventory at the DCs in each week to help predict future order quantity. This approach also rests on the theoretical foundation from the OM literature that a supplier's order quantity decisions can be improved when demand and inventory data are shared within the supply chain [18, 123].

Product Information. Aside from the EDI 852 data, we also collected additional informa-

tion of each drug's ATC code from public databases. ATC code is a classification system segmenting the drugs into distinct groups based on their chemical, pharmacological, and therapeutic properties. ATC code has five levels, with progressively more detailed information about the drug. We focus on the first level classification, also known as the main anatomical group, in which the drugs are divided into 14 main groups. Specifically, we first extracted the non-proprietary name for each NDC from the National Drug Code Directory, and then searched for the respective ATC code from the database at WHO collaborating center for drug statistic methodology. Since ATC code is widely used in pharma industry to classify drugs based on their characteristics, we later use ATC code as one of the grouping schemes in our forecasting models. We also obtained wholesale acquisition cost (WAC), which is the list price for each NDC. Real prices typically include discounts and rebates, but WAC is a widely accepted reference price [123].

Supply Chain Structure Information. The 133 NDCs are sold to 28 trade partners (including the top three wholesale distributors which represent 85% of the total annual U.S. sales) through their respective 247 DCs. We later tested at the TP level to see whether such supply chain information benefits the forecasts. On average, each trade partner purchases 64 NDCs from the drug manufacturer, while each DC receives roughly 34 NDCs every week. We provide the descriptive statistics of our dataset below.

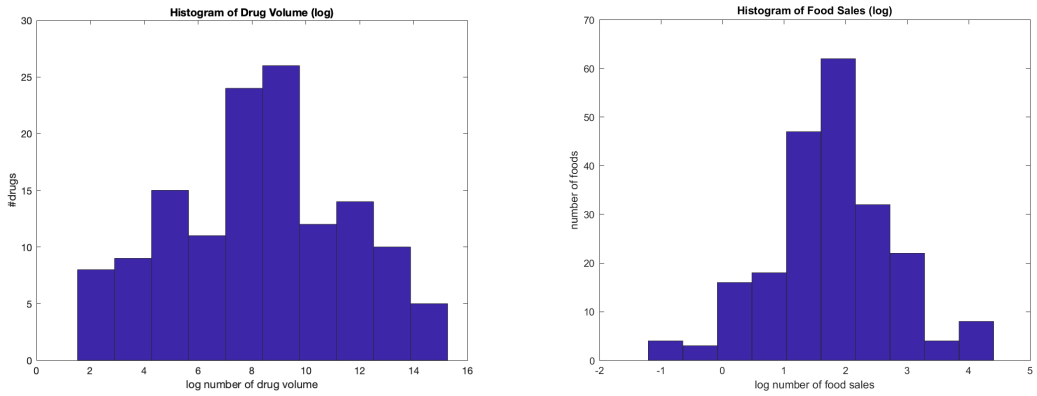
Table 3.1: Descriptive statistics of our dataset

	Mean	Std.	Min	Median	Max
# of DCs per TP	9	15	1	2	50
# of NDCs per TP	64	41	2	68	127
# of Observations per TP	123,031	272,775	240	25,749	970,393
Avg. Order Qty per TP	21,616	63,999	39	903	285,866

3.2 Retail Sales Dataset

Our retail dataset is extracted from the Walmart good sales dataset in the M5 competition [36]. The competition dataset contains daily sales and prices for thirty thousands of prod-

ucts from Walmart stores in California, Texas, and Wisconsin between 2011 and 2016. The products cover food, hobbies and household goods at 10 stores in each region. Retail dataset is used to verify the generalizability of cross-series learning. We select foods at Walmart stores in California to test our framework. There are many similarities between the food sales and pharma demands, for example, they are both under the influence of promotion, seasons, special events, etc. On the other hand, the sales range of food is much smaller than the demand range of drugs. Figure 3.5 shows the histograms of the drug demand volume and food sales volume in log scale. Food is more perishable, so food sales change faster than drug demand in a short period.



(a) Histogram of (log) drug volume (b) Histogram of (log) food sales

Figure 3.5: Ranges of drug demand and food sales

To facilitate comparison with pharma datasets, we preprocess the competition dataset to produce our retail dataset with similar format. We accumulate the daily sales of a product in 10 stores each week as the weekly sales of the product in that region and use the average of daily price in each week as the weekly price. Finally, the retail dataset contains weekly sales and prices of 216 food products at Walmart stores in California from January 2011 to April 2016. Every food product has 274 weeks of sales and price records. The statistic of foods' average weekly sales and prices in the retail datasets is shown in the table 3.2.

Table 3.2: Descriptive statistics of food weekly sales and prices

	Mean	Media.	Std.	Min	Max
sales	13.24	8.37	14.73	0	83.53
prices	3.23	2.51	1.95	0.97	11.35

3.2.1 Retail Sales

An example of a food product sales from 2011 to 2016 is shown in the Figure 3.6. Like pharma demand, due to the influence of hidden market factors such as periodic market activities and special events, retail sales also have spike values at certain time points and multiple seasonal patterns. ARIMA and ETS are often used to learn the seasonality of time series, such as additive seasonality and multiplicative seasonality. However, these seasonalities are not adequate to describe the complex seasonal patterns of sales. Therefore, we need to build more sophisticated models on large amounts of data.

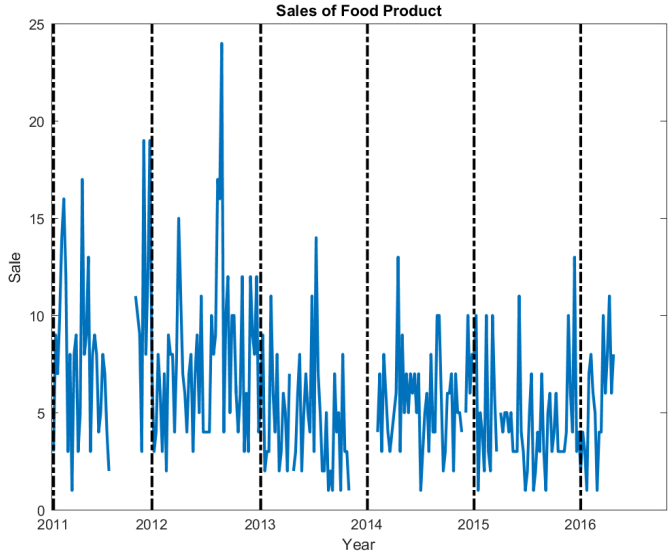


Figure 3.6: Retail Sales

Neural networks are universal estimators of functions [31, 39, 50], capable of modeling complex and non-linear patterns including seasonality [79, 111]. However, Nelson et al. [85] noted that the mathematical proof for the feasibility of neural networks for sea-

sonality modeling is only valid when there is no upper limit on the number of neurons. In practical forecasting applications, limited data availability may not support large-scale neural networks that can capture seasonal patterns. Later, our cross-series learning is used to address this limitation.

3.2.2 Sparsity of Retail Sales

The retail sales dataset includes sporadic zero sales due to the intermittency of the records. Even though weekly sales accumulates the daily sales of different stores within a week, many zeros are still left. Figure 3.7 shows the sparsity of all food product sales. Each row in the figure indicates whether there were non-zero sales in the week from 2011 to 2016. Blue pixels represent zero sales, and green represent non-zero sales. There are 59,184 observations in the retail sales dataset, but 33% of them are zeros. According to our preliminary test, building machine learning models on each such time series, the performance of advanced models (e.g., Neural Networks) is comparable to or even worse than the simple models (e.g., Linear regression).

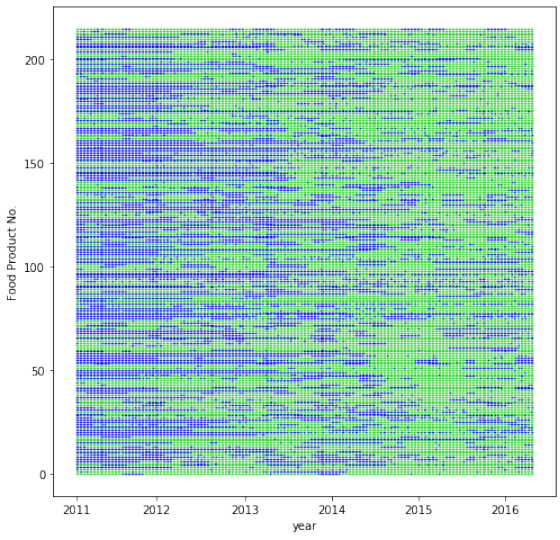


Figure 3.7: Sparsity of the retail sales

In Chapter 4, extensive numerical experiments are conducted on the first pharma

dataset. The observations and insights obtained from the first pharma dataset are validated on the second pharma dataset. We will use the retail sales dataset to verify the generalizability of the cross-series learning technique and the effectiveness of our generic forecasting framework in Chapter 5.

Chapter 4

Cross-Series Learning For Pharma Demand Forecasting

Accurate demand forecasting is the basis for supply chain efficiency since it essentially drives all important operational decisions, from raw material supply planning, production planning, inventory management, to financial goals. For a pharmaceutical (pharma) manufacturer, demand forecasting can be even more critical because (1) any mismatch between demand and supply could ripple through the drug distribution channel and impact the patients, sometimes even causing life-threatening situations; and (2) any demand that is not fulfilled could potentially lead to permanent lost sales from a patient, because patients who cannot afford the uncertainty in their order fulfillment may switch to an alternative drug. For drugs treating chronic illnesses, this could mean huge financial losses for the drug manufacturer.

Current demand forecasting in the pharma industry focuses on using simple statistical methods with historical demand to extract future demand patterns. However, we know that demand is under the influences of many factors, many times hidden factors, in addition to historical trends. Some of these factors apply across industries, such as the general economic environment, while others are unique to the pharma industry, such as distinctive demand patterns due to special pharma situations (e.g., investment buy-

ing), the change of regulations (pharma industry is highly-regulated), market competition between brands as well as between brand and generics, special contracts, and media effects (high public attention to pharma). Capturing such factors require more sophisticated models built upon a large amount of data, as well as domain knowledge of the industry. Indeed, the emerging concept of demand sensing, which focuses on identifying and including various factors affecting demand aside from historical demand, has attracted much attention [23, 92]. However, little has been done in practice, especially in the pharma industry. Simple time-series models often ignore these hidden factors or assume that these factors manifest themselves in the *individual* drug demand time series so that the future demand for a pharma product is simply a function of its own previous demands.

At the same time, in the pharma industry, FFS and EDI have generated a significant amount of data. This data, however, has not, in general, been utilized for demand forecasting or production planning [101]. While theoretical work from the operations management literature has shown potential value of downstream's demand and inventory data when being incorporated in the upstream's optimal production/inventory decisions [18, 123], it remains to be shown empirically whether the aforementioned EDI data can provide any additional value in the upstream's demand forecasting that we focus on. Further, since EDI data is unstructured, the next immediate question is how to effectively mine this additional information and adequately capture the hidden factors mentioned above in improving demand forecast.

Machine learning has been known to be an effective method to detect unknown patterns in structured and unstructured data. Recently, there have been more and more applications of machine learning in supply chain and operations management literature. It is well-known that effectively training and testing these machine learning models require a large amount of data for better estimation of the model parameters in order to achieve better performance. Yet, this condition is not readily met in most forecasting context since there exists a temporal constraint, i.e., old data from distant past may have little value to

the current prediction task (this is in fact confirmed in our data analysis).

The limitation of not being able to expand vertically to the past motivates us to look horizontally across drugs to increase the amount of data available to improve the forecast accuracy. However, it is not straightforward to determine how many and which drugs to include when learning across drugs to balance the tradeoff between the sample size and sample quality. Hence, we explore three different grouping schemes to enhance performance of the forecasting models. The first scheme uses a product segmentation approach commonly used in the industry based on demand volume and volatility. The second scheme uses product-based characteristics based on pharma domain knowledge. The third scheme requires no knowledge of the data and uses a time-series clustering algorithm to group the drugs. In addition, to investigate the value of downstream data, we also include inventory data as well as the information of the supply chain structure to explore whether and how much such information would help improve the model performance.

To execute our model framework, we work a large datasets from a top drug manufacturer whose name is hidden for confidentiality. The dataset includes weekly demand and inventory information extracted from EDI 852 over a 10-year period (2007-2017) for 133 unique products of this manufacturer represented as 133 NDCs, that are sold to 28 trade partners (e.g., wholesalers), through their respective 247 distribution centers (DC). Using this data, we develop a forecasting framework and various cross-drug training models that combine machine learning with pharma domain knowledge to predict future demand for each drug. The numerous design considerations pertaining to our framework (i.e., different machine learning algorithms to use, how to group drugs together, different levels of aggregation of the data based on supply chain structure, how much time lags of historical demand as well as inventory data is used) result in an extensive set of numerical experiments.

In Section 4.1, we develop the cross-drug forecasting framework, describe the detailed process for group drugs, introduce benchmarks, and discuss suitable forecasting

models and implementation details to be used with the data. In Section 4.2, we report our results on the benefit of cross-drug training, the benefit of grouping, the value of downstream inventory information, and the value of supply chain structure information. We also include robustness checks in terms of the models used, forecasting horizons, and the benefits of our model framework on inventory performance. In Section 4.3, we provide possible explanations for the evident effectiveness of RNN based on domain knowledge and additional numerical analyses. Section 4.4 concludes the observations and insights.

4.1 Model Development

Recall that each NDC has unique characteristics such as active ingredients, dosage form, route of administration, etc. Thus, there should exist no meaningful correlation structures between most drugs' time-series, and one cannot expect to be able to predict future demands for one drug using past demands of other drugs. Yet, exploratory analysis (such as the demand spikes in Figure 3.3) suggests that drugs may have some similar demand patterns. Therefore, we design to leverage their combined data to train a *single* model to capture these patterns. In what follows, we develop our model framework. Section 4.1.1 introduces notations for our demand forecast models and provides the general form of our cross-drug forecast model. Section 4.1.2 proposes our grouping schemes for cross-drug training. Section 4.1.3 discusses the choice of model (VAR and various machine learning models) and proposes to use recurrent neural network. Section 4.1.4 presents baseline models and Section 4.1.5 provides the implementation details.

4.1.1 Cross-drug Training

The objective of our model is to predict the *total* demand at the manufacturer from all DCs for each of the 133 NDCs . We first introduce the following important notations for the forecasting models.

- $\mathcal{I}$: set of all NDCs in the dataset, i.e, $\mathcal{I} = \{1, \ldots, I\}$.
- $\mathcal{J}$: set of all distribution centers in the dataset, i.e, $\mathcal{J} = \{1, \ldots, J\}$.
- $\mathcal{K}$: set of all trade partners in the dataset, i.e, $\mathcal{K} = \{1, \ldots, K\}$. Further, the set of all DCs from trade partner k will be denoted as $\mathcal{J}_k$, $k \in \mathcal{K}$, and we have $\sum_{k \in \mathcal{K}} J_k = J$.
- $x_{i,j,t}$: order quantity of drug $i \in \mathcal{I}$ from distribution center $j \in \mathcal{J}$ to manufacturer at time t
- $y_{i,j,t}$: inventory of drug $i \in \mathcal{I}$ in distribution center $j \in \mathcal{J}$ at time t
- X_{it}: cumulative order quantity of drug $i \in \mathcal{I}$ from all DCs to manufacturer at time t, i.e., $X_{it} = \sum_{j \in \mathcal{J}} x_{i,j,t}$
- Y_{it}: cumulative inventory of drug $i \in \mathcal{I}$ across all DCs at time t, i.e., $Y_{it} = \sum_{j \in \mathcal{J}} y_{i,j,t}$
- p, q: number of time period lags for order quantity and inventory, respectively, i.e, the most recent p weeks of order quantity and q weeks of inventory will be used in the forecasting model
- h: forecast horizon, measured in weeks
- $\hat{X}_{i,t+h}$: predicted cumulative order quantity for drug $i \in \mathcal{I}$ at time $t+h$, made at time t, i.e., predict at time t the demand in h periods in the future

When including the past p weeks of order quantities of all drugs in set $\mathcal{I}$ and q weeks of corresponding downstream inventory information, we design a cross-all-drug training model as follows

$$\hat{X}_{i,t+h} = f(X_{i,t}, \cdots, X_{i,t-p+1}, Y_{i,t}, \cdots, Y_{i,t-q+1}), \; \forall i \in \mathcal{I} \tag{4.1}$$

where the forecast of drug i is obtained from drug i's features only, but the mapping f is learned from data of all drugs. While learning across drugs greatly increases the

sample size hence provides a solution to the lack of data issue, we must also consider the tradeoff between sample size and sample quality for advanced models: learning and training across more "similar" drugs would potentially bring better sample quality. Therefore, a more advanced cross-training model would first group drugs according to some schemes, and build a forecasting model for these drugs cross-trained within each group. This idea will be further explored below.

4.1.2 Grouping Schemes

The key question in cross-drug training is which drugs should be trained and predicted together. In this section, we propose three grouping schemes based on different rationales including domain knowledge of the pharma industry.

Grouping by Demand Volume and Volatility The first scheme is based on demand volume (average order quantity per NDC) and demand volatility (measured by coefficient of variation - CV). Industry has also used these two criteria to group drugs in various situations (e.g., for product segmentation). Using the medians of demand volume and volatility, we partition all NDCs into four non-overlapping groups using the respective medians, namely, high volume-low volatility (HL), high volume-high volatility (HH), low volume-low volatility (LL) and low volume-high volatility (LH). Summary statistics regarding order quantities (EU) of each group are summarized in Table 4.1.

Table 4.1: Order quantities (EU) in four groups based on volume/volatility

Group Name	Mean	Median	CV.	Min	Max	# of NDCs	# of Obser.
HL	393,084	29,160	3.13	0	25,086,100	55	2,434,628
HH	179,289	3,784.04	3.22	0	9,064,080	11	167,734
LL	3,124	1,200	1.91	0	145,920	11	354,526
LH	1,704	0	5.18	0	514,967	56	487,987

With these four groups, we build four models, one for each group. Each model is trained using only the information from the drugs belonging to that specific group. The

model is shown in Eq. (4.2) as follows

$$\hat{X}^{(g)}_{i,t+h} = f^{VV}(X^{(g)}_{i,t}, X^{(g)}_{i,t-1}, \cdots, X^{(g)}_{i,t-p+1}, Y^{(g)}_{i,t}, Y^{(g)}_{i,t-1}, \cdots, Y^{(g)}_{i,t-q+1}), g \in \{\mathsf{HL}, \mathsf{HH}, \mathsf{LL}, \mathsf{LH}\}, \tag{4.2}$$

where $X^{(g)}_{i,t}$ and $Y^{(g)}_{i,t}$ stand for the order quantity and inventory information of drug i in group g at time t, respectively, that our model will include. The forecast of drug i is still obtained using drug i's features but the mapping f^{VV} (with "VV" for volume and volatility) is now learned using the data of the drugs in the same demand volume and volatility group.

Grouping by ATC code The second way we propose to group the drugs is through ATC code. Recall that drugs in the same ATC code have similar therapeutic, pharmacological, and chemical properties. Thus, we can think of ATC Code as a product-characteristics criteria while volume and volatility as a product-demand criteria to segment the drugs. Drugs in our dataset belong to 6 major ATC code groups, namely, A, B, C, G, J and N, where A refers to Alimentary tract and metabolism drugs, B refers to blood and blood forming organs, C refers to cardiovascular system, G refers to genito-urinary system and sex hormones, J refers to anti-infectives for systemic use, and N refers to nervous system [88]. Summary statistics of order quantities from NDCs in each ATC code group are provided in Table 4.2. Observe that some ATC groups have more NDCs and observations than others. The cross-drug training model by ATC code can be expressed as

$$\hat{X}^{(\phi)}_{i,t+h} = f^{A}(X^{(\phi)}_{i,t}, X^{(\phi)}_{i,t-1}, \cdots, X^{(\phi)}_{i,t-p+1}, Y^{(\phi)}_{i,t}, Y^{(\phi)}_{i,t-1}, \cdots, Y^{(\phi)}_{i,t-q+1}), \phi \in \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{G}, \mathsf{J}, \mathsf{N}\}, \tag{4.3}$$

where $X^{(\phi)}_{i,t}$ and $Y^{(\phi)}_{i,t}$ stand for the order quantity and inventory information of drug i at time t, respectively, with ATC code ϕ. Now, the mapping f^{A} (with "A" for ATC codes) is learned using the data of drugs in the same ATC code group.

Grouping by clustering algorithm So far, we have proposed a demand-based approach

Table 4.2: Order quantities (EU) in ATC code groups

Group Name	Mean	Median	CV.	Min	Max	# of NDCs	# of Obser.
A	72,223	7,200	2.66	0	2,930,385	44	568,582
B	416,259	4,881	2.07	0	6,558,360	12	344,521
C	137,728	6,443	2.31	0	4,741,848	18	840,984
G	2,351,915	130,375	2.07	0	25,086,100	7	234,391
J	31,472	3,600	2.59	0	857,400	26	64,229
N	88,855	2,100	4.08	0	9,064,080	17	768,342

recognized by industry and a product-based domain knowledge approach to create features to group NDCs for cross-drug training. If such domain knowledge is not readily available, we propose to use clustering algorithms. Here, we use an unsupervised machine learning technique called K-means clustering on the drugs historical demand to group the NDCs into K different clusters so that drugs in the same group are more similar to each other than to those in other groups. In particular, we adopt the dynamic time warping (DTW) algorithm [9] to measure the similarity between two demand time-series. Essentially, DTW finds the optimal alignment between two drugs' demand time series, and thereby measures their shape similarity accordingly. Due to the magnitude differences in the order quantities of different drugs, we normalize each time series (by subtracting it by the respective mean and then dividing by the standard deviation) prior to DTW computation.

Selecting the optimal number of clusters of drugs, K, requires a balance between the clustering quality and the number of observations in each cluster. To do so, we use the Davies-Bouldin index (DBI) as the clustering evaluation metric, defined as

$$DBI = \frac{1}{K}\sum_{i=1}^{K} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right), \tag{4.4}$$

where σ_i and σ_j are the intra-cluster distances of clusters c_i and c_j, respectively. Note that an intra-cluster distance measures the average DTW distances of all pairs of drugs' demand time series within the same cluster. In contrast, the inter-cluster distance $d(c_i, c_j)$

measures the distance between the two clusters c_i and c_j, that is, the average DTW distances between all pairs of drugs' demand time series in which one is selected from cluster c_i and the other is selected from c_j. The number of clusters K is selected so that it has the lowest value of DBI, which indicates better quality for the respective clustering performance. For our data, our approach results in five clusters. Table 4.3 provides the summary statistics of order quantities of drugs in each cluster.

Table 4.3: Order quantities (EU) in the generated clusters

Cluster Index	Mean	Median	CV.	Min	Max	# of NDCs	# of Obser.
1	225,470	4,000	6.42	0	25,086,100	20	851,298
2	961	0	11.19	0	514,967	18	86,908
3	161,985	10,274	2.52	0	5,469,480	32	1,137,336
4	3,744	46	3.91	0	453,570	27	392,932
5	529,288	61,560	2.17	0	13,720,620	36	976,401

The cross-drug training model by clustering can be expressed as

$$\hat{X}_{i,t+h}^{(\kappa)} = f^C(X_{i,t}^{(\kappa)}, X_{i,t-1}^{(\kappa)}, \cdots, X_{i,t-p+1}^{(\kappa)}, Y_{i,t}^{(\kappa)}, Y_{i,t-1}^{(\kappa)}, \cdots, Y_{i,t-q+1}^{(\kappa)}), \kappa \in \{1, 2, \ldots, K\}, \tag{4.5}$$

where $X_{i,t}^{(\kappa)}$ and $Y_{i,t}^{(\kappa)}$ stand for the order quantity and inventory information of drug i at time t in cluster κ, respectively. The mapping f^C (with "C" for clustering) is learned using the data of drugs in the same cluster.

4.1.3 Machine Learning Models

With the groups obtained from the preceding schemes, we next develop the learning model to predict future drugs' demand in each group. As discussed, vector auto-regression (VAR), a method used for multiple time-series forecasting, is not suitable in our problem because in practice, it is recommended to use on a small number of time-series that are correlated with each other [52]. This practical guideline clearly limits the applicability of VAR to our dataset. Indeed, if we build a model to predict demand for 10 drugs using 8

lags (2 months of data), there are 81 coefficients per VAR equation, giving a total of 810 coefficients to be estimated. Further, as noted before, VAR only captures linear relationships.

We propose to use a ML algorithm to forecast drugs' demand in a group. Since the data exhibits non-linear patterns, we focus the discussion on non-linear methods, while linear methods will be used as one of the baseline models (see Section 4.1.4). Among a plethora of non-linear methods, there are three widely-used classes in the literature: support vector regression (SVR), random forest (RF), and neural networks. A basic structure for neural networks is the fully connected neural network (FC). Next, we will give a short description of these methods.

Linear Regression (LR) is the simplest machine learning algorithm to capture linear patterns from the data. Compared with ES, which assumes the influence of historical observations to future variable decays exponentially over time, linear regression is more flexible. Furthermore, linear regression can incorporate different types of data, but requires the mapping f (in Section 4.1.2) to be linear in the inputs. In our case, the response variable is the prediction of drug order quantity, and the inputs are the historical order quantities and possibly inventory information at the DCs. For example, if we want to use drug i's past p weeks order quantity to forecast its next week order quantity, the linear model L_i is shown in Equation (4.6):

$$L_i : \hat{X}_{i,t+1} = \beta_{i,1} X_{i,t} + \beta_{i,2} X_{i,t-1} + \cdots + \beta_{i,p} X_{i,t-p+1} \tag{4.6}$$

where $\beta_i = (\beta_{i,1}, \beta_{i,2}, \cdots, \beta_{i,p})$ are parameters or weights for each of the p periods. Both the ES and linear models can only recognize linear patterns in time series. To further explore nonlinear patterns, we need to use nonlinear models.

Random Forests (RF) is an ensemble learning method for both classification and regression tasks. It constructs many *decision trees* at training time since a combination of learning models help increase the overall result. Decision tree is a popular machine

learning algorithm which can fit complex dataset. A decision tree starts from a root node which includes all the training data. If the standard deviation of the data in the current node is larger than the threshold, it builds a decision boundary for a feature. The boundary splits the data into two subgroups and saves the subgroups in the child nodes. This procedure is performed recursively for each node until the standard deviation of the data in the node is lower than the threshold. Note that random forest is essentially a bagging algorithm, it builds a large collection of trees and then average them, to help reduce the variance of the estimated prediction function. To further prevent overfitting, tree pruning can also be used to remove tree leads with high errors and complexity.

Support Vector Regression (SVR) is a non-parametric technique based on kernel function. Using different implicit mappings with various kernel functions, one can transform our data into the dual space, and find an optimum function to fit the data pattern. Equation (4.7) shows how to apply SVR to our time series data.

$$\hat{X}_{i,t+1} = \sum_{n=1}^{N} (\alpha_n - \alpha_n^*) G([X_t^{(n)}, \cdots, X_{t-p+1}^{(n)}], [X_{i,t}, \cdots, X_{i,t-p+1}]) \tag{4.7}$$

where $X_t^{(n)}$ is the nth training data at time t, α_n and α_n^* are two non-negative multipliers for the nth training data, function $G(.)$ is the kernel function. Linear kernel, Gaussian kernel and Polynomial kernel are widely used kernel functions.

Artificial neural networks (ANN) have been used in time series forecasting problems due to their effectiveness in capturing nonlinear patterns (see Hill et al., 1996 and the references therein). In this dissertation, we explore two popular architecture of ANNs, including a traditional fully connected neural network (FC) and a more modern structure called recurrent neural network (RNN).

Fully Connected Neural Network (FC) is the basic architecture of ANNs with great learning ability suited for many kinds of applications. In addition, FC without hidden layers is reduced to the learning process of linear models. The input of our neural network is a

vector:

$$[X_{i,t}, X_{i,t-1}, \cdots, X_{i,t-p+1}, Y_{i,t}, Y_{i,t-1}, \cdots, Y_{i,t-q+1}]$$

and the output is a scalar, which is the prediction of the order quantity $\hat{X}_{i,t+1}$. The structure of FC is shown in Figure 4.1. FC is also known as a feed forward neural network, indicating the one-way flow of data from the previous layer to the next layer.

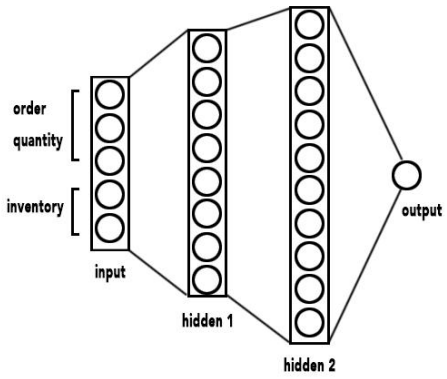


Figure 4.1: Fully connected neural network

To learn demand patterns such as the demand spikes with SVR and RF, the timing of the spikes needs to be treated as categorical variables. However, mixed data of numerical and categorical data usually hurt the performance of both SVM and RF. In contrast, a **Recurrent Neural Network (RNN)**, a special type of neural network with memory cells to enable tracking of short and long-term dependencies in the input, can potentially capture hidden patterns well in our data. Note that RNN is particularly suitable for processing sequential data, e.g., time-series data [48], hence making it a great candidate. The structure of RNN is schematically shown in Figure 4.2.

Each computation unit of RNN is called a cell. Different RNNs may have different cell structures (LSTM, GRU, etc.), but their most crucial feature is that each cell's outputs have connections backward. Therefore, at each time step, the cell receives inputs as well as its own output from the previous time step. As a result, a cell's outputs are influenced not only by the most recent input, but also by the entire history of past inputs. Further,

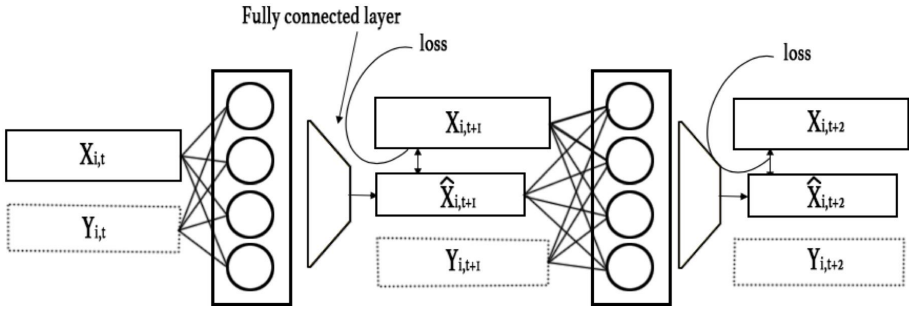


Figure 4.2: Recurrent neural network

each cell implements a series of gates in which information can be passed on or forgotten. This particular architecture makes it possible for RNN to explore the temporal dynamic information from a time series [60]. In our demand forecasting context, the input of the cell at period t is the order quantity and inventory at period t as well as the output of the cell at period $t-1$. The output is the prediction of order quantity at time t.

4.1.4 Baseline/benchmark models

Unfortunately we cannot obtain the company's internal forecast (it is missing from the data shared with us). Hence, we select the baseline based on the reported benchmarks in the literature (Table 1), the current practice in the industry (Section 1.1), and the results from the mentioned questionnaire about the current demand forecasting practices (Section 1.2). These sources converge to the forecasting methods of moving average (MA), linear regression (LR), and simple exponential smoothing (ES). Hence, to be inclusive and conservative, we include all three (MA, LR, ES) and choose the best out of the three as our baseline models. Further, to make sure that the state-of-the-art baseline models are used, we improve the aforementioned simple baseline models (e.g., ES models) by utilizing the innovative state space model developed by Hyndman and Athanasopoulos [52], which include 30 separate models. Note that these sophisticated models were developed to automatically forecast demands for thousands of drugs. Each model has an observation equation and transition equations, one for each state (level, trend, seasonal)

with additive or multiplicative errors. The *ETS forecast package* in R, implementing the above model, automatically chooses the most appropriate ES method as well as the optimal parameters for the forecasting task. ETS model empirically provides slightly more accurate forecasts than ARIMA [52], which is hence not selected as one of the benchmarks.

While the aforementioned baseline models help us understand the value of our proposed cross-drug forecasting approach compared to the current demand forecasting practices in the pharma industry, we also would like to validate the performance and robustness of RNN, compared to other machine learning algorithms, i.e., linear regression (LR), support vector regression (SVR), and random forrest (RF). Hence, we report the performances of these models as well for the completeness of the study, as robustness checks.

4.1.5 Implementation Details

Time lags. To estimate the future order quantity, we use past weeks' information (e.g, the previous p weeks' historical demand to the manufacturer and/or q weeks' inventory information). We test different time lags $p, q = 1, 2, \ldots$ and choose the best performing combination of p and q values for each model via cross-validation. For instance, whenever a model uses inventory information, we test all combinations of time lags for order quantity ($p = 1, 2, \ldots, 10$ weeks) and inventory information ($q = 0, 1, \ldots, 10$ weeks) and report the best one. Thus, in total, we test 110 combinations for each model.

Time-series Cross-validation. To ensure a model robustness across different time periods, its performance is measured using the time-series cross-validation technique [52]. In particular, we use three consecutive years as the training set and the fourth year as the test set. Within each training set, we perform cross-validation by using a rolling forecasting origin. That is, we use the observations in a rolling window to train the model and the observation outside the window to validate. In each cross-validation round, we compute the accuracy metrics and select the model parameters that achieve the best

performance on the test set. This time-series cross-validation procedure for model evaluation is suited to time-series data since serial correlation and potential non-stationarity may exist [8]. When available, we also include p weeks of data before the start of the training set time to ensure all the training samples will have the right inputs. For example, suppose the training set is from 2008 to 2010. We include in the training set the last p weeks' order quantities at the end of 2007 to predict the order quantity for the first week in 2008. Generation of the training and test set is demonstrated in Figure 4.3.

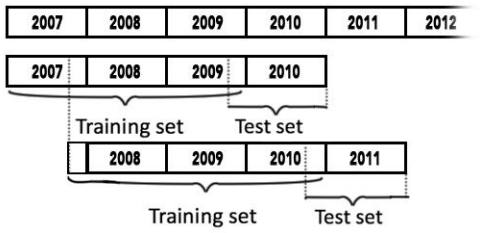


Figure 4.3: Training set and test set

Using the processing methods described above, we create a data matrix for each training and test set. Since we have records of I drugs in a given three years, and there are k weeks in these three years, we want to use the past p weeks of order quantity and q weeks of inventory data to predict the order quantity for the next week. As a result, we obtain an $Ik \times (p+q+1)$ matrix. The first column of this matrix contains the order quantity to be predicted, labeled as the response variable.

Parameter Tuning. In RNN, there are several hyperparameters to be tuned, e.g., the number of neurons, the number of layers, learning rate, batch size, etc. We also need to decide the suitable number of lags used in the model. This can be done using the standard grid-search and cross validation methods outlined above. At the end of this process, we will obtain the optimal set of hyperparameters to forecast demand. In particular, p and q can vary from 1 to 10, the number of neurons ranging from 100 to 1000 in each cell, initial learning rate ranging from 1e-4 to 1e-1, batch size ranging from 2^4 to 2^6, the number of epochs ranging from 100 to 300. Regarding the optimizer for RNN, we use

the adaptive moment estimation (Adam) optimization algorithm, which is known to work well in most practical applications, with the exponential decay rate for the first moment estimates ranging from 0.9 to 0.999, and the second moment estimate set at the default value of 0.999. The L_2 regularization parameter ranges from 10^{-3} to 10^{-1}.

Evaluation Metrics. Note that the popular mean absolute percentage error (MAPE) cannot be used for our dataset due to the possibility of zero demands. Thus, we use normalized mean square error (NMSE), normalized mean absolute error (NMAE) and bias to evaluate our models' performance. Normalization is used to facilitate comparison across different models and different NDCs. For a thorough review of forecast accuracy measures [53]. In reporting the forecasting accuracy, for a given forecasting horizon (e.g., 1 week), we evaluate the forecasting accuracy for each group by taking the average across all drugs in that group. For most of the models we report results for forecasting horizon of 1 week, with the robustness check of the forecasting horizon varying from 1-8 weeks in Section 4.2.6.

4.2 Results and Discussion

This section reports results of our proposed framework in terms of benefit of cross-drug forecasting, benefit of grouping drugs, value of inventory information, value of supply chain structure information, and their implications in the pharma demand forecasting context. We then conclude with a few robustness checks.

4.2.1 Performance of Baseline Models

As discussed, the selected baseline models represent the state-of-the-art forecasting approach in the pharma industry. In this section, we first compare their performance against the company's internal forecasts, which are only available from January 2012 to March 2016. Table 4.4 presents the performance measures, along with the respective 95% con-

fidence intervals, of the internal forecasts in comparison with the baseline models, i.e., moving average (MA), exponential state-space models (ES), and linear regression (LR).

Table 4.4: Forecasting bias and accuracy measures of the baseline models vs. internal forecasts

	MA	ES	LR	Internal Forecast
NMSE	1.58 ± 0.15	2.08 ± 0.18	1.63 ± 0.16	4.10 ± 0.24
NMAE	0.29 ± 0.05	0.34 ± 0.06	0.30 ± 0.05	0.45 ± 0.06
bias	0.00 ± 0.01	0.02 ± 0.02	-0.04 ± 0.00	0.09 ± 0.04

The numerical results in Table 4.4 clearly show that the baseline models clearly outperform the company's internal forecasts across different accuracy measures. This also confirms the soundness in our (conservative) choice of the baseline models. Due to the superior performance of the baseline models against the company's internal forecasts, for the remainder of the computational study, we only report the performance of the baseline models.

4.2.2 Benefit of Cross-drug Forecasting

Table 4.5 presents the performance measures of the cross-all-drugs training models in comparison with the baseline models, i.e., moving average (MA), exponential state-space models (ES), and linear regression (LR). None of the baseline models use cross-drug training, i.e., it builds one model for each drug.

Table 4.5: Forecasting bias and accuracy measures of cross training models using all drugs

	Baselines (No Cross-drug)			With Cross-drug Training					
	MA	ES	LR	VAR	LR	SVR	RF	FC	RNN
NMSE	1.71 ± 0.09	2.06 ± 0.15	1.67 ± 0.10	2.74 ± 0.23	1.58 ± 0.09	1.83 ± 0.11	1.60 ± 0.10	1.50 ± 0.09	0.98 ± 0.12
NMAE	0.31 ± 0.03	0.34 ± 0.04	0.30 ± 0.03	0.43 ± 0.08	0.30 ± 0.03	0.32 ± 0.05	0.29 ± 0.04	0.29 ± 0.03	0.25 ± 0.04
Bias	0.03 ± 0.02	-0.02 ± 0.03	-0.03 ± 0.02	-0.06 ± 0.06	-0.04 ± 0.02	0.03 ± 0.06	0.01 ± 0.04	0.02 ± 0.02	0.01 ± 0.04

The numerical results in Table 4.5 show that, except for VAR, cross-drug training is beneficial across all performance metrics, and RNN performs the best. In general, cross-drug training with more advanced models gives better results. In contrast, using

cross-training on all drugs with just simple models like LR, or even SVR do not give much improvement. With RNN's performance significantly exceeding that of the linear models, this result indicates that RNN is able to pick up some nonlinear patterns. The other nonlinear models do not perform well on the dataset, possibly because RNN is better suited for time-series data [48], as we mentioned. Indeed, our analysis later shows that RNN is the only method that can efficiently capture demand spikes (Section 4.3). As previously discussed, VAR is not suitable for the cross-drug training we propose to do. Indeed, its performance is significantly worse than even those of the baseline models (without cross-training); and thus, in the remaining analysis, VAR is not further considered.

While cross-all-drugs forecasting increases the sample size, hence brings good improvements for some ML models, we next see how grouping the drugs and building a separate model for each group (i.e., decrease the sample size for each model while increasing the sample quality) could further improve the forecasting performances using ML models.

4.2.3 Benefit of Grouping Drugs

In reporting the results, the suffix "4" indicates grouping by the four volume/volatility categories, "ATC" by ATC code, and "DTW" by clustering. To facilitate comparison between competing methods, we only report the best performing baseline model's performance measures and percentage improvement of cross-drug training models over that of the best baseline. We use a dash whenever a model is worse than the baseline. Further, for easy exposition, we only report the performance of LR and RNN and move those of other ML models (which are all inferior to RNN) to Appendix. For brevity, we also move "bias" to Appendix.

Tables 4.6 and 4.7 show the benefit of grouping by demand volume/volatility and by ATC, respectively. The tables show that the performance of RNN is significantly better than that of LR and in general, both LR and RNN benefit from grouping. For example,

improvement in NMSE over the baseline for all drugs has increased from 5.4% to 7.2% for linear models, and from 41.3% to 46.7% for RNN when grouping by demand volume/volatility. Further, only RNN achieves consistent improvement for all drug groups, especially for the low volume drugs (Table 4.6). For different ATC code groups, RNN has significant improvement for most of the groups. Even for the ATC group with smaller sample size (group J), training within ATC groups shows improvement of 10.8%. This result suggests that, in practice, if we can segment the drugs with similar proprieties with a sufficient sample size, cross-product training using ATC code can offer great benefit.

Table 4.6: Improvement of cross-drug training models with grouping by volume/volatility

		Best Baseline	LR	LR_4	RNN	RNN_4
NMSE	All Drugs	1.67	5.4%	7.2%	41.3%	46.7%
	HL	1.05	3.8%	3.8%	39.0%	44.8%
	HH	1.44	16.7%	5.6%	44.4%	47.2%
	LL	2.69	9.3%	10.0%	--	29.4%
	LH	17.37	--	--	--	49.8%
NMAE	All Drugs	0.30	0.0%	0.0%	16.7%	16.7%
	HL	0.29	3.4%	3.4%	10.3%	17.2%
	HH	0.37	10.8%	2.7%	27.0%	24.3%
	LL	0.69	4.3%	5.8%	--	18.8%
	LH	0.81	--	--	--	19.8%

Table 4.7: Improvement of cross-drug training models with grouping by ATC code

		Best Baseline	LR	LR_ATC	RNN	RNN_ATC
NMSE	All Drugs	1.67	5.4%	6.0%	41.3%	47.3%
	A	1.46	6.2%	4.8%	2.7%	24.0%
	B	0.55	14.5%	12.7%	38.2%	52.7%
	C	1.24	--	--	--	4.0%
	G	1.45	--	--	35.9%	37.9%
	J	1.20	10.0%	8.3%	--	10.8%
	N	7.53	15.1%	3.6%	39.2%	42.1%
NMAE	All Drugs	0.30	0.0%	3.3%	16.7%	16.7%
	A	0.39	5.1%	5.1%	2.6%	5.1%
	B	0.29	10.3%	10.3%	17.2%	27.6%
	C	0.35	--	0.0%	--	0.0%
	G	0.39	--	--	12.8%	15.4%
	J	0.36	5.6%	2.8%	--	13.9%
	N	0.42	9.5%	2.4%	19.0%	16.7%

Table 4.8 reports and compares different models' performance under all three group-

ing schemes. While both LR and RNN show benefits from using any grouping schemes, RNN has superior performance. In particular, using clustering, over all drugs, RNN has achieved 53% improvement in NMSE over the baseline compared to 41.3% improvement when no grouping is used. Generally, all grouping schemes improves RNN performance across different groups. In addition, while the performance of cross-drug training models using different grouping schemes are generally comparable to each other for the high-volume products, grouping helps RNN to particularly improve the low-volume products, whose demands are typically very difficult to forecast. For low-volume products, grouping by demand volume/variance performs the best, followed by grouping by clustering, then ATC code.

Table 4.8: Improvement of cross-drug training models with grouping schemes

		Best Baseline	LR	LR_4	LR_ATC	LR_DTW	RNN	RNN_4	RNN_ATC	RNN_DTW
NMSE	All Drugs	1.67	5.4%	7.2%	6.0%	7.8%	41.3%	46.7%	47.3%	53.3%
	HL	1.05	3.8%	3.8%	2.9%	3.8%	39.0%	44.8%	44.8%	50.5%
	HH	1.44	16.7%	5.6%	4.9%	13.2%	44.4%	47.2%	54.9%	57.6%
	LL	2.69	9.3%	10.0%	--	5.9%	--	29.4%	--	9.7%
	LH	17.37	--	--	--	--	--	49.8%	25.4%	27.1%
NMAE	All Drugs	0.30	0.0%	0.0%	3.3%	3.3%	16.7%	16.7%	16.7%	20.0%
	HL	0.29	3.4%	3.4%	3.4%	3.4%	10.3%	17.2%	17.2%	20.7%
	HH	0.37	10.8%	2.7%	5.4%	8.1%	27.0%	24.3%	29.7%	27.0%
	LL	0.69	4.3%	5.8%	--	2.9%	--	18.8%	--	13.0%
	LH	0.81	--	--	--	0.0%	--	19.8%	12.3%	11.1%

In summary, the reported results demonstrate that learning across drugs that are more similar to each other helps to better detect common trends and patterns shared by the drugs in that group. In other words, it is more effective when we build a machine learning model for each group with more similar drugs and sufficient size. While grouping drugs by volume/volatility or by ATC codes has better interpretability, grouping by clustering does not require any knowledge of the data and also performs well.

4.2.4 Value of Downstream Inventory Information

This section reports the value of downstream inventory information in manufacturer's demand forecasting. Specifically, Table 4.9 shows RNN's performance for models using and not using inventory information for various grouping schemes. For performance of LR and the other nonlinear models, we refer to Table 8 in Appendix. Generally, adding inventory information almost always improves RNN's performance on all groups regardless of the grouping schemes, except for the low volume and high volatility drugs when using clustering. It is particularly helpful when grouping by demand volume and volatility. In addition, we find that among the many lags we tried for the inventory information, $q = 1$ typically provides the best results. In other words, inventory information in the distant past does not help, i.e., including the inventory information in the most recent period is sufficient to garner most benefit.

Table 4.9: Improvement of cross-drug training models by using inventory information

		Best Baseline	RNN	RNN_inv	RNN_4	RNN_4_inv	RNN_ATC	RNN_ATC_inv	RNN_DTW	RNN_DTW_inv
NMSE	All Drugs	1.67	41.3%	46.1%	46.7%	49.1%	47.3%	51.5%	53.3%	55.7%
	HL	1.05	39.0%	43.8%	44.8%	47.6%	44.8%	49.5%	50.5%	53.3%
	HH	1.44	44.4%	47.2%	47.2%	47.9%	54.9%	56.3%	57.6%	60.4%
	LL	2.69	--	--	29.4%	61.0%	--	--	9.7%	14.1%
	LH	17.37	--	--	49.8%	51.5%	25.4%	37.1%	27.1%	--
NMAE	All Drugs	0.30	16.7%	16.7%	16.7%	20.0%	16.7%	20.0%	20.0%	20.0%
	HL	0.29	10.3%	17.2%	17.2%	20.7%	17.2%	20.7%	20.7%	20.7%
	HH	0.37	27.0%	18.9%	24.3%	27.0%	29.7%	32.4%	27.0%	27.0%
	LL	0.69	--	--	18.8%	26.1%	--	--	13.0%	2.9%
	LH	0.81	--	--	19.8%	23.5%	12.3%	45.7%	11.1%	--

4.2.5 Value of Supply Chain Structure Information

The goal of this study is to forecast the manufacturer's demand for each drug. So far, we do so by forecasting the aggregate demand from all DCs from all trade partners (TP). Another approach is to forecast demand at the TP-level or DC-level and then aggregate forecasts to obtain the total demand required for each drug at the manufacturer. With

this approach, not only do we learn from other drugs, we could also learn from other DCs or learn amongst the group of DCs belonging to the same TP. One can argue that this forecasting approach can be beneficial because all DCs from the same TP must share some similarities such as ordering patterns. However, predicting at the downstream level also sees more volatility. Thus, it remains a question whether the benefit will overcome the drawback. This approach falls under the category of group time series forecasting, which is of significant interest to many researchers [51]. Thus, this section explores this forecasting paradigm and provides detailed discussion on the findings.

At DC-level, a model for DC j with input consisting of last p weeks' order quantity and q weeks' inventory is shown in Eq. (4.8) below

$$\hat{x}_{i,j,t+h} = f^{DC}(x_{i,j,t}, \cdots, x_{i,j,t-p+1}, y_{i,j,t}, \cdots, y_{i,j,t-q+1}), \tag{4.8}$$

where the mapping f^{DC} is learned using the data of drugs at DC j. Note that some DCs may not have sufficient training data, i.e., rank deficiency problem. For the DC-level model, given there are 247 DCs, we should have 247 models, one for each DC. However, as mentioned, due to the amount of data at the DC or TP level, the more complex machine learning models can suffer from overfitting. Thus, we could only build DC-level model using linear regressions. In particular, when building the DC-level model, we first train a model across all DCs, dubbed LR_DC. If a specific DC is rank deficient, we will use the predictions obtained from the LR_DC model for that DC.

To train the TP-level model, we use cross training among all DCs from the same trade partner. The rationale is that DCs from the same trade partner share the similar supply chain management system; hence, the ordering policies and inventory control of these DCs are more likely to follow similar patterns. If DC j belongs to trade partner φ, we can use Eq. (4.9) with input of last p weeks' order quantity and q weeks' inventory to make

predictions as follows

$$\hat{x}^{(\varphi)}_{i,j,t+h} = f^{TP}(x^{(\varphi)}_{i,j,t}, \cdots, x^{(\varphi)}_{i,j,t-p+1}, y^{(\varphi)}_{i,j,t}, \cdots, y^{(\varphi)}_{i,j,t-q+1}), \tag{4.9}$$

where $x^{(\varphi)}_{i,j,t}$ and $y^{(\varphi)}_{i,j,t}$ are the order quantity and inventory of distribution center j which belongs to trade partner φ at time t.

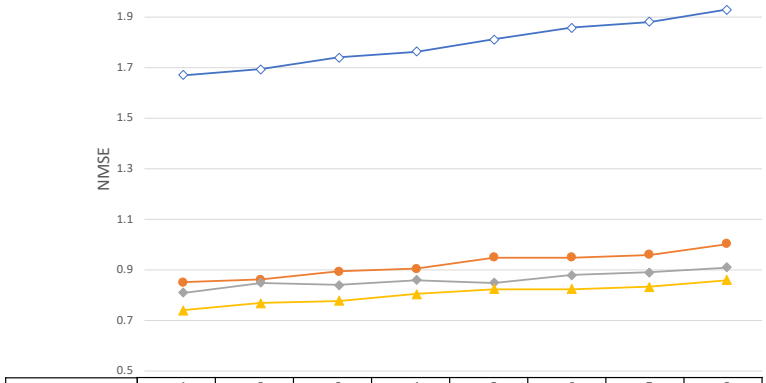
Tables 5-7 in Appendix compare the performances of the three different levels of models: the aggregate level, the TP level and the DC level. The results confirm that in general, the performances of the models at the TP or DC level are no better than that from the aggregate model across all metrics (NMSE, NMAE and bias). Given the amount of additional effort it requires to run each TP-level or DC-level model, forecasting at these levels may not be worthwhile unless each TP has a large number of DCs and each DC has a large amount of data.

4.2.6 Robustness Check

Robustness of RNN's performance. So far, we explored four machine learning models (LR, SVR, RF, FC) compared to RNN, in combination with three different grouping schemes to predict future drug demands, with and without cross-drug information. All findings confirm that RNN has significantly better performance in terms of its forecast accuracy than the other ML models (see Appendix for complete results).

Robustness of forecasting horizon. Forecast horizon refers to how far in the future we predict the demand. Figure 4.4 shows that, as expected, the forecast accuracy for all models decreases as we forecast further in the future (from 1 week to 8 weeks) because of higher uncertainty. This trend continues as the forecasting horizon goes beyond 8 weeks. However, regardless of the grouping schemes, cross-drug training with downstream inventory information consistently leads to significant forecast improvements for all horizons.

Impact of RNN forecasting models on inventory performance. To further validate the



	1	2	3	4	5	6	7	8
Best Baseline	1.67	1.69	1.74	1.76	1.81	1.86	1.88	1.93
RNN_4_inv	0.85	0.86	0.89	0.90	0.95	0.95	0.96	1.00
RNN_ATC_inv	0.81	0.85	0.84	0.86	0.85	0.88	0.89	0.91
RNN_DTW_inv	0.74	0.77	0.78	0.80	0.82	0.82	0.83	0.86

Figure 4.4: NMSE of Cross Training Models with RNN Over Prediction Horizon

benefits of our forecast models, we roughly estimate its respective service level (measured by the average number of weeks with stockout per year), the average annual stockout costs, and the average annual inventory cost (when the order-up-to policy is used for inventory replenishment). The inventory cost computes the total value of inventory across drugs. Table 4.10 shows that RNN with different grouping schemes significantly outperforms the baseline in terms of service level, stockout and inventory cost.

Table 4.10: Service level and inventory cost of RNN with different grouping strategy

	Best Baseline	RNN	RNN_inv	RNN_4	RNN_4_inv	RNN_ATC	RNN_ATC_inv	RNN_DTW	RNN_DTW_inv
Service Level (# weeks with stockouts per year)	4.71	1.69	1.33	2.04	2.08	2.64	1.37	2.64	2.56
Inventory Cost (in million $)	3.42	3.54	3.43	2.56	2.49	2.76	3.08	2.76	2.66
Stockout Cost (in million $)	0.17	0.14	0.11	0.13	0.11	0.11	0.12	0.12	0.13

4.3 Explanation of the Benefits of RNN

So far we have seen that RNN consistently outperforms other ML methods (LR is the simplest ML method). In this section, we try to provide some insights into the effectiveness of RNN over other methods for drug demand forecasting.

First, previous explorations of drugs' demand patterns suggest that many drugs demonstrate demand spikes in January, June, and December (see Figure 4.5). Specifically, a demand is considered to be a spike if it is three standard deviations above the annual average demand. This phenomenon is closely related to the prevalent investment buying behavior of the pharma distributors that is well-documented and studied in the literature [101, 123]. Specifically, investment buying refers to the phenomenon where distributors intentionally purchase large quantities of pharma products in anticipation of manufacturers' price increases in order to make profits by speculation on inventory. While exact price increase dates for different drugs are uncertain, these price increases often occur at the beginning, end, and/or in the middle of the year. Such timing may also reflect the manufacturer's incentive to get rid of inventory to meet financial/sales targets at certain times of the year. Given the prevalence of such phenomena, a good forecasting model should be able to capture such spikes.

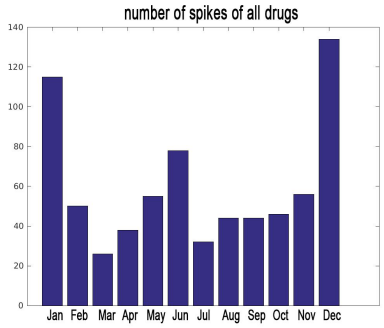


Figure 4.5: Timing of Demand Spikes from Our Data

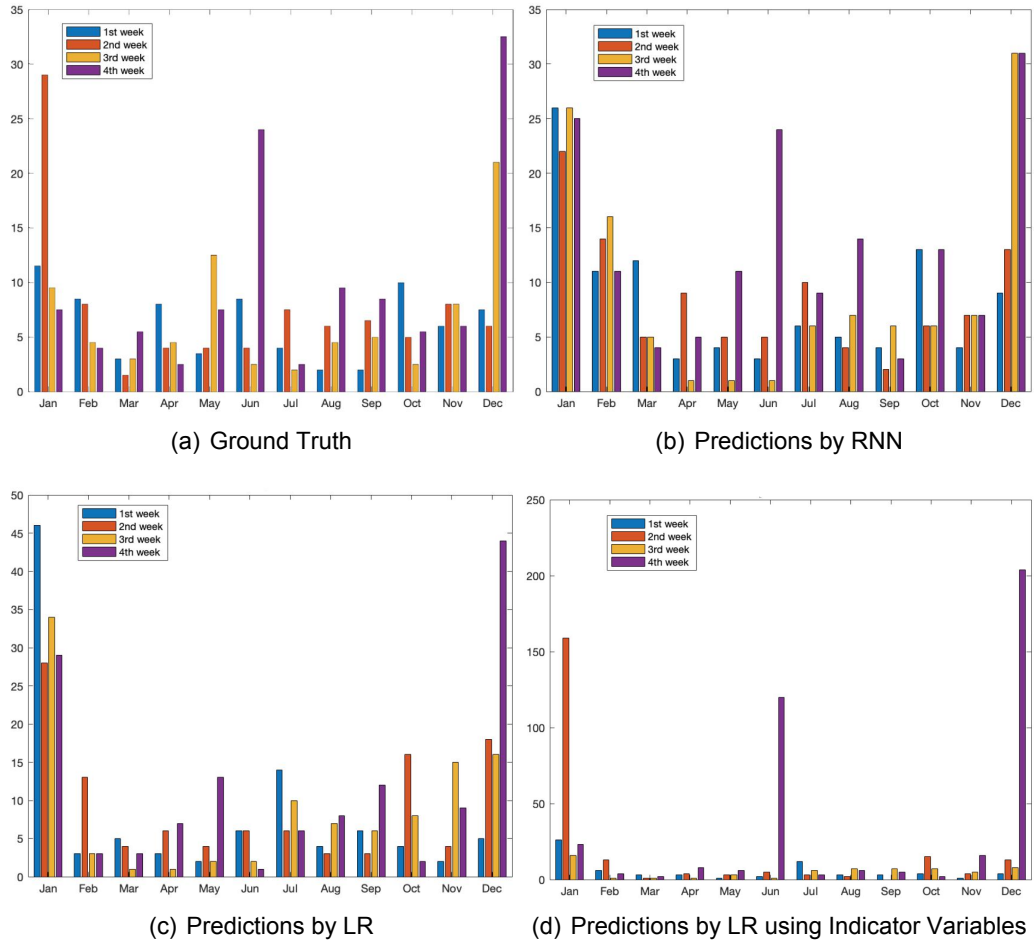
RNN is capable of capturing temporal feature, e.g., demand spikes, due to its special design with memory cells that can remember distant past. Figure 4.6 shows the weekly spikes distribution in different months from the data and from the preditions using RNN and LR. If there are five weeks in a month, we merge the fifth week spikes into the fourth week. Observe that demand spikes' pattern generated from RNN's predictions matches well with the ground truth. In using LR, we tried two models: with and without indicator variables, which are added manually to capture the demand spikes in Jan, June and December.

While LR's predictions without indicator variables cannot capture the demand spikes in June, LR with indicator variables is too aggressive in the way that many non-spikes values are mistaken for spikes, which leads to its poor performance. On the other hand, RNN predictions "mimic" the ground truth more closely.

Second, RNN outperforms the other models due to its ability to generate rather complicated non-linear features through its hidden layers. To empirically validate this observation, we purposely help the competing methods (LR, SVR, RF and FC) with feature engineering. That is, we extract new features from existing data and use them as extra inputs to enhance performance of these models. These features include exponential moving averages of historical order quantities, the minimum, maximum, variance, maximum absolute deviation around the mean of order quantities, and the linear slope of past order quantities computed for respective window sizes from 2 to 10. In total, there are about 50 newly created features. However, even with the inclusion of engineered non-linear features, other models cannot achieve similar performance as RNN. This experiment confirms that RNN can capture many hidden demand patterns which are missed by other methods. This helps to explain the better performance of RNN.

4.4 Conclusion

Demand forecasting drives many operational decisions and directly relates to companies' financial goals. Demand forecasting in the pharma industry is especially critical to drug manufacturers due to the unique features of the industry. However, the performance of existing forecasting models in the pharma industry seems to have reached a bottleneck, often limited by the amount of available data. At the same time, the availability of supply chain channel data and the rise of machine learning technologies provide new opportunities. Under this situation, in this dissertation, we propose a new forecasting framework which leverages information across drugs regarding historical demand, and non-demand channel information such as downstream inventory data as well as supply chain struc-



(a) Ground Truth (b) Predictions by RNN

(c) Predictions by LR (d) Predictions by LR using Indicator Variables

Figure 4.6: Spiked patterns captured by different forecasting models

ture information in demand forecasting. This framework not only helps us to gain a large amount of data that allows for more complex machine learning models, but also proposes various grouping schemes to guarantee sample quality for cross-drug training to capture common hidden factors affecting demand, hence improving the forecasting accuracy. Further, while analytical work has long shown the value of downstream inventory information, we are the first to empirically capture such value in the cross-drug demand forecast setting.

Using the dataset from the top drug manufacturer, we conducted extensive computational experiments to test our proposed forecasting framework. Our results provide some

important insights:

- Training across drugs indeed improves demand forecasting accuracy, showing significant benefits in forecast accuracy compared to the baseline.
- Cross-drug training is most effective for demand forecasting of low-volume drugs, whose forecasts are the most difficult in practice, possibly because it helps the most to alleviate the problem of the lacking of data.
- Cross-drug training with different grouping schemes based on product-specific information, either by demand volume/volatility or by product-based domain knowledge (ATC code in our case), is effective. On the other hand, clustering algorithms can be a great option when lacking domain knowledge in grouping.
- RNN consistently performs the best, far exceeding the others ML methods because (1) it can most effectively capture hidden factors such as demand spikes caused by investment buying behavior and (2) its special architecture makes it suitable for time-series data. While the latter is reported in many other studies in the literature, we are the first to document the former in the pharma context.
- Downstream inventory information is indeed beneficial in demand forecasting; however, as expected, any distant past inventory information does not bring additional benefit. This finding empirically confirms the value of downstream inventory information as shown in the operations management literature, but also complements that with more practical guidelines of what product groups' inventory information to collect.
- While more detailed supply chain structure information such as the downstream DC-level or TP-level data is helpful to learn across DCs or TPs to capture possible common hidden factors, its benefits do not seem to overcome the loss of accuracy due to disaggregation. As a result, it may not be worthwhile to collect the more detailed DC-level data for the aggregate demand forecasting.

Our proposed forecasting framework (including grouping schemes, using downstream inventory and supply chain structure information, in combination with the RNN models) can be applied to other pharma manufacturers, wholesalers, and possibly other industry based on its robust performances. Domain knowledge is important for making modifications to this framework when adapting to other industries. This industry-specific customization could be a promising research direction since products in different industries may have unique characteristics that can be extracted and incorporated into the forecasting framework to fully boost its performance. Finally, as leading pharma companies (Pfizer, Sanofi, etc.) have already considered using AI platforms to help the drug development process, these companies might also benefit from using AI applications in their pharma supply chains (e.g., demand forecasting), as suggested by this research.

Chapter 5

Generalizability of Cross-Series Learning

The findings in Chapter 4 demonstrate that learning across similar time-series can be very suitable for pharma demand forecasting. We can summarize the learning process across products in pharma settings as follows: drug product attributes (e.g., ATC code) were used to group time series and downstream inventory and supply chain structure information was used to assist in demand forecasting. Compared with benchmark models, our cross-series learning models achieved significant improvements for our pharma dataset. In order to verify the generalizability of the cross-series methods to other industries, this chapter aims to develop a generic framework that includes the entire cross-series learning process from data preprocessing, time series grouping to model training, testing, and validation. First, we reproduced our experiments on the first pharma dataset, then validated our findings on the second pharma dataset, and finally confirmed the effectiveness of the framework on a retail sales dataset.

In Section 5.1, we provide a detailed description of the proposed forecasting framework. This framework is then tested on the second pharma dataset and the retail dataset, and the obtained experimental results are presented in Section 5.2. Finally, a summary of our observations is given in Section 5.3.

5.1 Generic Cross-Series Learning Framework

In our forecasting framework, there are four steps from reading in the raw time series data to obtaining the prediction results. First, it is necessary to perform data preprocessing operations and generate synchronized features for each time series data. Then, according to different grouping schemes, we divide the global time series into subgroups containing similar time series. Subsequently, we construct a machine learning model for each group of time series, use a rolling forecast origin to validate performance and find optimal hyperparameters through grid search. Finally, we postprocess the predictions generated by the well trained models. The structure of our framework is illustrated in Figure 5.1.

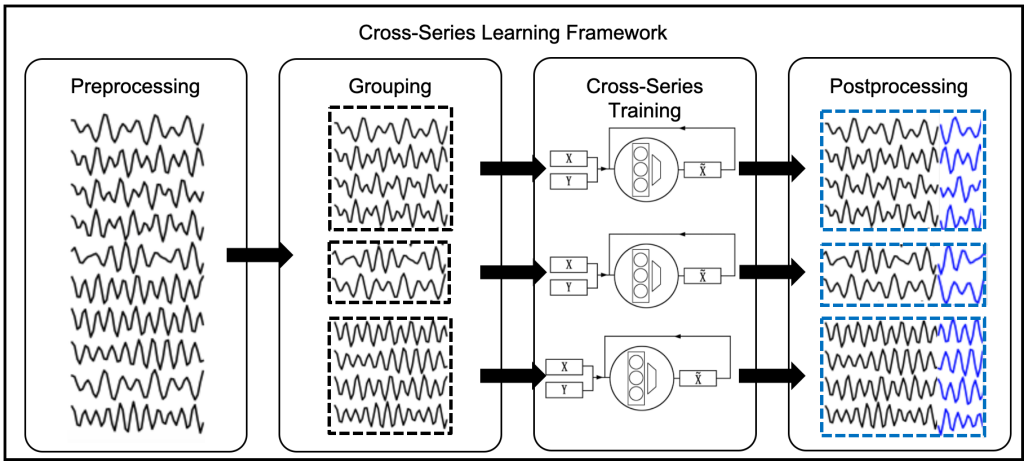


Figure 5.1: Generic Cross-Series Learning Framework

Each step in the above process corresponds to a component in the framework. Next, the operations included in each component are detailed.

5.1.1 Data Preprocessing

Raw time series data needs preprocessing before being used for training. The preprocessing operations are different for various types of time series and applications. Our framework implements some common operations including data selection, normalization,

and deseasonalisation. The introduction and generation of additional features can also be conducted at this stage.

- *Data Selection*

  In practical applications, the underlying patterns often vary with time. Distant past data is of little use for current predictions, and it may also distract the focus of learning from the most recent patterns. Therefore, we try to select the latest data as training data, similar to the drug demand forecasting (i.e., using the data of the last three years). For clearly categorized time series, we can roughly select the related time series without grouping schemes. For example, in the retail sales dataset, we can first choose similar products sold in the same region (e.g., food sold in CA), and then perform a more precise separation in the grouping stage.

- *Normalization*

  Since we are training on different time series and the value ranges of observations are not the same, normalization is necessary for some machine learning models (e.g., SVR). We use the mean value of the time series to normalize each observation, called mean-scale transformation. Equation (5.1) shows this operation.

$$\tilde{x}_{i,t} = \frac{x_{i,t}}{\frac{1}{T}\sum_{\tau=1}^{T} x_{i,\tau}} \tag{5.1}$$

  where $x_{i,t}$ indicates the observation at time t of ith time series, T is the total number of time points, and $\tilde{x}_{i,t}$ stands for the corresponding normalized observation.

  It is worth noting that normalization does not always improve performance because it may wash out features related to scale, such as drug volume in the pharma demand forecasting problem. Hence, normalization is an optional operation that should be determined according to the model's performance.

- *Deseasonalisation*

  According to the results of M competition [77], researchers found that machine learning models trained with deseasonalized data can generate more accurate predictions. Nelson et al. [47] also compared the performance of neural networks trained with and without deseasonalized time series data and concluded that neural networks also benefit from deseasonalisation. However, studies show that neural networks are universal estimators of functions [31, 39, 50], capable of modeling complex and non-linear patterns including seasonality [79, 111]. Moreover, in our pharma demand forecasting experiments, even without deseasonalisation, RNN models still manage to capture spikes at the beginning, middle, and end of the year, which proves the advantages of neural networks in modeling intricate seasonal patterns. Nelson et al. [85] discuss the necessity of time series deseasonalisation when using neural networks. They note that the mathematical proof for the feasibility of neural networks for seasonality modeling is only valid when there is no upper limit on the number of neurons. In practical forecasting applications, limited data availability may not support large-scale neural networks that can capture seasonal patterns. Deseasonalisation can also make neural networks focus on learning other patterns, such as trends and cyclic movements. Therefore, we implemented deseasonalisation operations in the preprocessing component by using the *Seasonal and Trend decomposition using Loess* (STL) methods developed by Cleveland et al. [29]. STL decompose a time series into seasonal, trend and residual components. Figure 5.2 shows an example of seasonal and trend decomposition of a weekly retail sales time series with the period equal to 4 (monthly seasonality).

  After decomposition, we recompose the trends and residuals as deseasonalized inputs. Since cross-series learning can help overcome the data limitation when constructed large scale neural networks, the benefits of deseasonalisation are not always obvious. Like normalization, the usage of deseasonalisation is also optional depending on the application and model performance.

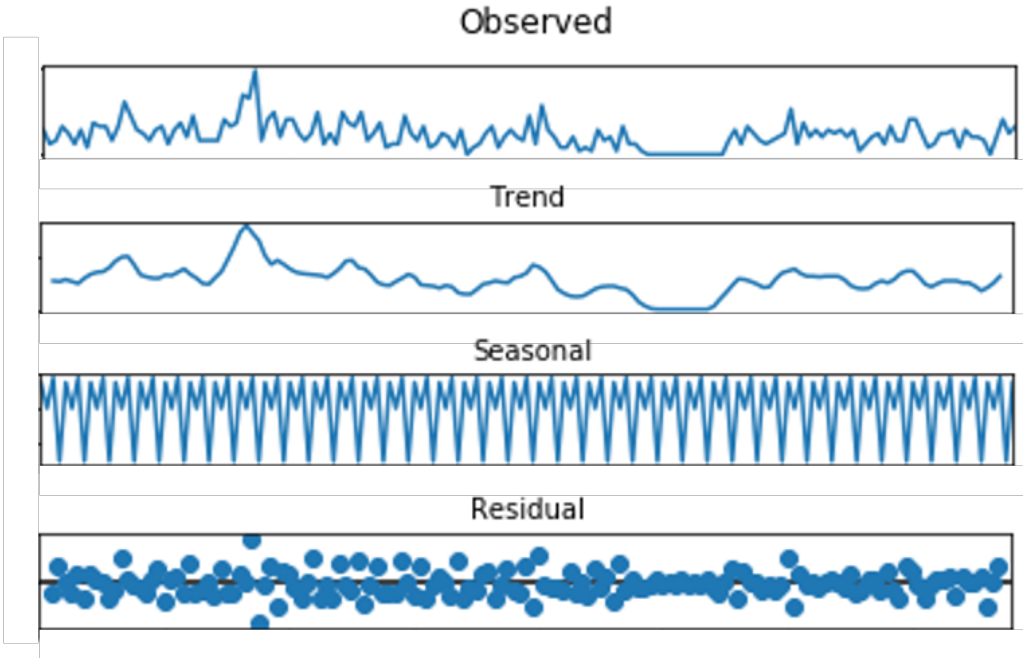


Figure 5.2: Seasonal and trend decomposition of retail sales

- *Additional Features Generation*

  For synchronized digital information (e.g., inventory information in the pharma demand dataset and price information in the retail sales dataset), we can import them directly as additional features. These features may also need preprocessing, such as normalization. For non-synchronized information, interpolation or sampling is necessary to maintain synchronization with the original time series. Non-digital information should be mapped to corresponding digital values, such as indicators for special events. If there is no additional information, the time series itself can be used to generate useful features. From the sub-series in a sliding window before each observation, features can be obtained such as exponential smoothing moving average, maximum value, minimum value, standard deviation, linear-gradient, coefficient of variation, etc. Seasonal components and trend components generated by STL can also be used as new features.

5.1.2 Time Series Grouping

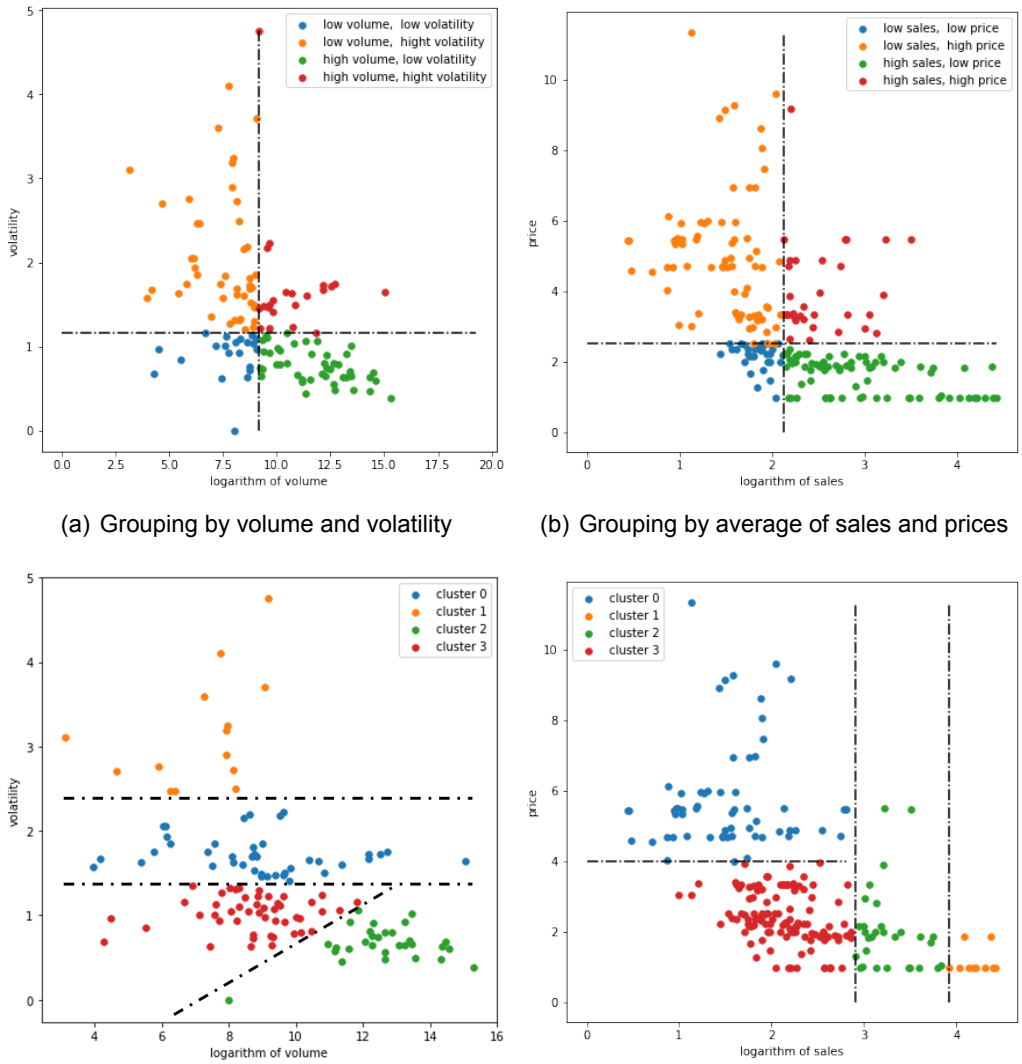
To avoid time series with widely different properties from interfering with each other and to balance the tradeoff between sample size and sample quality, the global time series is separated into subgroups containing similar time series. There are two types of metrics used to measure the similarity of time series. One is from the domain knowledge of corresponding applications, and the other is based on time series clustering. Both metrics have their advantages. Grouping schemes based on domain knowledge are more convenient, intuitive, and interpretable. Schemes based on time series clustering can be applied to a wide range of time series and do not require additional information. In the pharma demand forecasting experiments, two types of grouping schemes achieved comparable improvements, and accordingly, the type of grouping scheme depends on the model's actual performance.

- *Grouping with Domain Knowledge*

  Cross-series learning is based on the assumption that the related time series may have the same behavior patterns. In the product prediction environment, the most intuitive grouping standard is product property, such as medicines to treat the same disease, products of the same type, and clothing of the same style. If the product classification information is available in the data, for example the ATC code in the pharma demand dataset and the department id in the retail sales dataset, we can directly use them to group the time series in the data selection stage. Otherwise, there are conventional segmentation methods based on time series statistical features, including average, coefficient of variation, etc. For example, in Figure 5.3(a), drugs are grouped according to the order quantity volume (i.e., average) and volatility (i.e., coefficient of variation) and in Figure 5.3(b), food products are divided based on their average sales and prices.

- *Grouping with Time Series Clustering*

  For general prediction tasks that lack domain knowledge, we can group time series by



(a) Grouping by volume and volatility

(b) Grouping by average of sales and prices

(c) Clustering by volume and volatility

(d) Clustering by average of sales and prices

Figure 5.3: Grouping by time series statistical features

using clustering approaches. Specifically, we constructed a symmetric distance matrix whose elements correspond to the dissimilarity of each pair of time series sample and then input the distance matrix into the clustering algorithm. There are two approaches to compare the similarity between time series. The first approach is to generate a feature vector for each time series, that is, to map each time series to a point in the feature space [5]. Still taking the volume and volatility of drugs and the sales and prices of retail goods as examples, Figure 5.3(c) and Figure 5.3(d) show the clustering results of K-

means on pharma and food products. Grouping schemes based on clustering optimize the similarity of time series within the group, but interpretability is worse than schemes based on domain knowledge. Hyndman et al. [55] proposed features for capturing time series dynamics, including strength of linearity, strength of trends, strength of seasonality, first order of autocorellation, etc. Another approach is to use the nature of the time series itself instead of any handcrafted features. The most common method is to average the Euclidean distance of the observations at the corresponding positions between the two time series. Moreover, there are other time series for oriented distance, such as dynamic time warping distance (DTW) [95], Hausdorff distance [93], Gower distance [43], and symmetric segment-path distance (SSPD) [10]. One problem with the second approach is that the time series may contain long runs of missing values. Figure 5.4 illustrates our solution, which is to estimate the average of distance between the overlapping segments (i.e., red parts) in the time series.

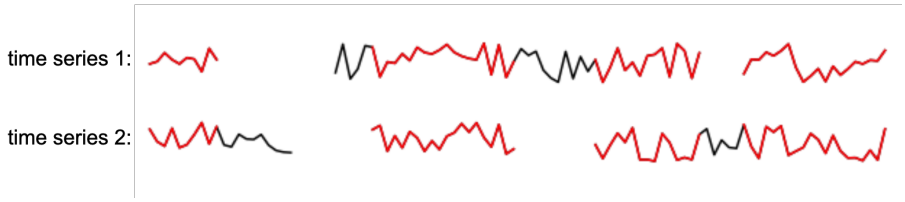


Figure 5.4: Estimate time series distance with missing values

In addition to K-means, we also use *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) [62] and *Hierarchy Clustering Analysis* (HCA) [91] clustering algorithms implemented by Matlab.

5.1.3 Cross-Series Training

After grouping, we start to train the machine learning models for each subgroup of the time series. First, we transform all inputs into a data matrix that meets the requirements of different types of machine learning models. Next, a machine learning model is built based on predefined hyperparameters. Finally, the best performing models are chosen

and used to generate predictions.

- *Generation of Data Matrix from Multiple Time Series*

  The method for generating the data matrix is a technique called *rolling forecast origin* [112]. Figure 5.5 depicts the procedure of using a rolling forecast origin to create a data matrix from multiple time series. As shown in the figure, two types of time series (i.e., white and blue) represent two input features. The first feature (i.e., white) is also the one to be predicted. The windows (i.e., the red frames) with different sizes roll along with the two types of time series respectively. The last position of each window is called the *forecast origin*. Each time a window moves, a sample at the current forecast origin is generated. Observations in the window are the predictor values, and the response value (i.e., grey and dark blue) is right after the forecast origin. By concatenating the predictor values generated from the two types of time series at the same forecast origin, we get the predictor matrix (i.e., X). The response values from the first time series constitute the response vector (i.e., y). Finally, the data matrices generated from the time series (i.e., TS1, TS2, etc.) in the same subgroup should be combined as the final data matrix.

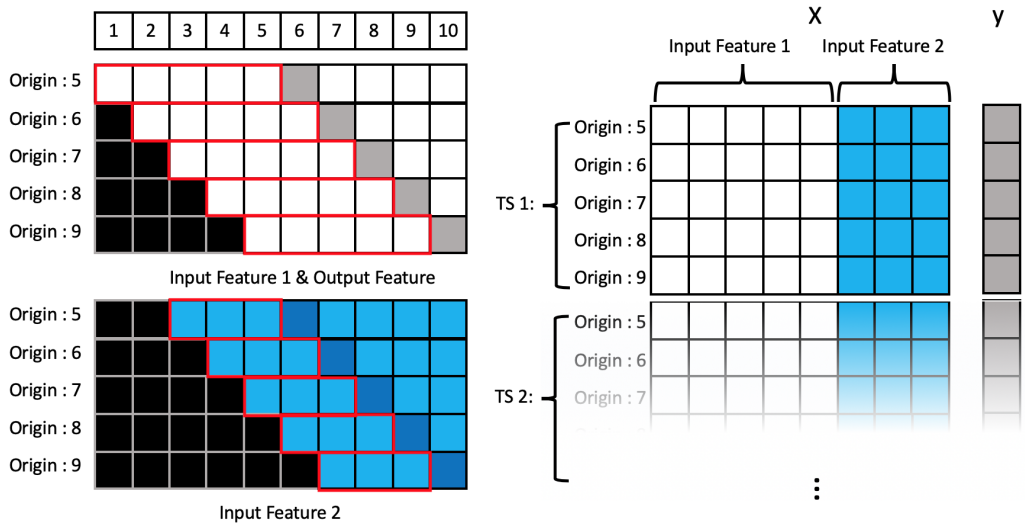


Figure 5.5: Rolling forecast origin

The data matrix generated in Figure 5.5 can be used as input for linear regression,

support vector regression, random forest, and multilayer perceptron. However, the input matrix of RNN uses another method to concatenate the samples with different features, as shown in Figure 5.6.

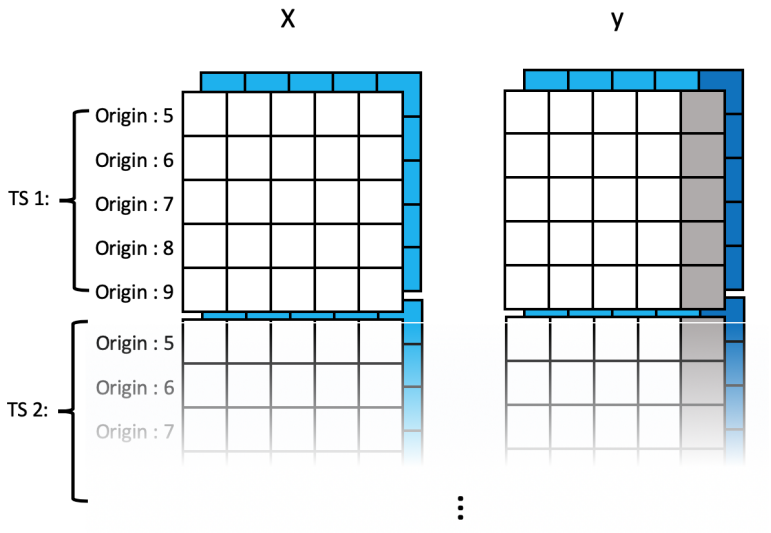


Figure 5.6: Data Matrix of RNN

- *Machine Learning Models and Hyperparameters*

  In the pharma demand forecasting problem, cross-series learning improves the prediction accuracy of machine learning models, including linear regression, support vector regression, random forest, fully connected neural network and recurrent neural network. In the generic framework, we formalize these machine learning models and the benchmark models (e.g., exponential smoothing and moving average) into a unified training process. First, a model is built based on predefined hyperparameters, then the data matrix is generated by rolling forecast origin to train the model, and finally, model performance is evaluated, and the trained model is saved for future use. The parameters shared by all the models are the number of time lags, forecast horizon, input features, output (predicted) feature, path to save the model, etc. Additionally, there are parameters for training neural networks, such as learning rate, batch size, number of epochs, etc. Table 5.1 lists the parts of the essential hyperparameters owned by each model

implemented in our framework. The open-source Python libraries that we used to implement each machine learning model are attached after each model's name.

Table 5.1: Important hyperparameters of different machine learning models

Machine Learning Model	Important Hyperparameters	Explanation
Exponential Smoothing (statsmodels.tsa.holtwinters)	trend	Trend type (additive, multiplicative)
	seasonal	Seasonal type (additive, multiplicative)
	seasonal_period	Season period
	damped	Use damped version or not (True or False)
Moving Average (statsmodels.tsa.api)	decay	Decay rate
VAR (statsmodels.tsa.api)	ridge	Ridge parameter
Linear Regression (sklearn.linear_model)	intercept	Use intercept or not (True or False)
	ridge	Ridge parameter
Support Vector Regression (sklearn.svm)	kernel	Type of kernel (e.g., rbf, sigmoid)
	degree	Degree of polynomial kernel
	gamma	Kernel coefficient
	C	Regularization parameter
Random Forest (sklearn.ensemble)	n_estimators	Number of Estimators
	max_depth	Maximum depth of the tree
	min_samples_split	Minimum number of samples to split a node
	min_samples_leaf	Minimum number of samples in a leaf
	bootstrap	Use bootstrap or not
Fully Connected Network (tensorflow)	n_hiddens	The list contains number of neurons on each hidden layer
RNN (tensorflow)	n_neurons	Number of neurons in each cell
	cell_type	Cell type (e.g., basic and LSTM)
	n_layers	Number of cell layers
	drop_rate	Dropout rate for dropout layer

- *Hyperparameters Tuning*

The parameter selection process is performance-driven. Model performance should be evaluated with various possible parameter combinations to obtain the optimal hyperparameters. An automated grid search pipeline was implemented for this laborious task: (1) write the tuning parameters and their ranges into a configure file, (2) recursively traverse all the parameter combinations, (3) build and train a model based on the parameters, (4) record the out of sample performance for the model, and (5) select the best performing model and corresponding parameters.

5.1.4 Postprocessing

Postprocessing mainly performs reverse preprocessing operations on the prediction results. For example, if we use deseasonalisation in the preprocessing stage, it is necessary to recompose the seasonal components with the predictions. Then the results should be converted back to the original scale if we apply the mean-scale transformation to the time series.

Our python library of generic cross-series learning framework is in github repository: *https://github.com/zxdan523/CrossSeriesForecast*.

5.2 Experiments and Results

5.2.1 Validation on the second pharma dataset

To further validate the performance of our proposed forecasting framework, we tested the proposed cross-drug forecasting models on a second dataset from a different manufacturer using the same forecasting model framework and cross-validation procedure. This dataset includes all transactions over the period from Jan 2011 to Dec 2017 between a drug manufacturer and its trade partners' DCs, collected weekly for 112 unique NDCs via 5 TPs and 73 DCs.

Table 5.2 clearly shows our framework is indeed beneficial, VAR is not a suitable method of choice, and RNN has the best performance across different accuracy metrics.

Table 5.2: Forecasting bias and accuracy measures of cross training models using all drugs (second dataset)

	Baselines			With Cross-drug Training					
	MA	ES	LR	VAR	LR	SVR	RF	FC	RNN
NMSE	4.67 ± 0.10	4.95 ± 0.14	4.29 ± 0.13	6.63 ± 0.22	3.98 ± 0.14	3.72 ± 0.13	3.67 ± 0.13	3.65 ± 0.11	2.69 ± 0.37
NMAE	0.42 ± 0.02	0.44 ± 0.02	0.41 ± 0.02	0.62 ± 0.05	0.40 ± 0.02	0.40 ± 0.02	0.38 ± 0.02	0.39 ± 0.02	0.30 ± 0.06
Bias	-0.01 ± 0.01	-0.14 ± 0.02	-0.11 ± 0.02	-0.20 ± 0.05	-0.07 ± 0.02	-0.05 ± 0.02	-0.07 ± 0.02	-0.04 ± 0.03	-0.01 ± 0.02

Table 5.3 confirms the superior performance of RNN across different grouping schemes (particularly grouping by demand volume/volatility and clustering), which is consistent with our previous observations in the first dataset. For the counterparts of information in Tables 4.6 and 4.7, refer to Appendix. Table 5.4 verifies the benefit of downstream inventory information in pharma demand forecasting.

Table 5.3: Improvement of cross-drug training models with grouping schemes (second dataset)

		Best Baseline	LR	LR_4	LR_ATC	LR_DTW	RNN	RNN_4	RNN_ATC	RNN_DTW
NMSE	All Drugs	4.29	7.23%	6.29%	1.17%	1.40%	37.30%	45.45%	43.36%	49.18%
	HL	1.34	8.21%	6.72%	1.49%	1.49%	39.55%	45.52%	44.03%	48.51%
	HH	4.05	0.49%	2.22%	--	1.98%	11.60%	46.42%	34.57%	49.14%
	LL	0.98	2.04%	2.04%	--	0.00%	--	65.31%	--	68.37%
	LH	5.12	14.06%	14.65%	15.23%	--	--	28.52%	--	28.32%
NMAE	All Drugs	0.41	2.44%	2.44%	0.00%	0.00%	26.83%	39.02%	26.83%	43.90%
	HL	0.38	2.63%	2.63%	0.00%	0.00%	34.21%	39.47%	31.58%	42.11%
	HH	0.54	0.00%	0.00%	--	3.70%	1.85%	37.04%	16.67%	40.74%
	LL	0.53	0.00%	1.89%	--	0.00%	--	52.83%	--	56.60%
	LH	0.59	0.00%	5.08%	8.47%	--	--	37.29%	--	37.29%

Table 5.4: Improvement of cross-drug training models by using inventory information (second dataset)

		Best Baseline	RNN	RNN_inv	RNN_4	RNN_4_inv	RNN_ATC	RNN_ATC_inv	RNN_DTW	RNN_DTW_inv
NMSE	All Drugs	4.29	37.30%	50.12%	45.45%	60.61%	43.36%	42.89%	49.18%	67.60%
	HL	1.34	39.55%	51.49%	45.52%	61.94%	44.03%	44.78%	48.51%	54.48%
	HH	4.05	11.60%	37.28%	46.42%	49.63%	34.57%	29.14%	49.14%	59.51%
	LL	0.98	--	--	65.31%	33.67%	--	--	68.37%	70.41%
	LH	5.12	--	--	28.52%	24.41%	--	--	28.32%	22.46%
NMAE	All Drugs	0.41	26.83%	36.59%	39.02%	41.46%	26.83%	26.83%	43.90%	46.34%
	HL	0.38	34.21%	42.11%	39.47%	42.11%	31.58%	42.11%	42.11%	44.74%
	HH	0.54	1.85%	18.52%	37.04%	37.04%	16.67%	35.19%	40.74%	46.30%
	LL	0.53	--	--	52.83%	15.09%	--	--	56.60%	58.49%
	LH	0.59	--	--	37.29%	25.42%	--	--	37.29%	35.59%

The above results show that our model framework works well and major insights hold true to the second dataset. This, together with our discussion and justification of the models we choose (Section 4.1) as well as the possible explanation of the effectiveness of the RNN models (Section 4.3), provides evidence of the generalizability of our model framework and results.

5.2.2 Validation on the retail dataset

To ensure that the cross-series learning method can be extended to other industrial products, we test our generic framework on Walmart retail sales dataset. The dataset contains weekly sales and weekly prices of 216 food products from 10 Walmart stores in California from 2011 to 2016. The generic framework completes the entire process from data preprocessing, model training to performance evaluation.

In the preprocessing stage, we use three consecutive years data as the training set and the fourth year data as the test set, as we did with the pharma demand dataset in Section 4.1.5. Therefore, there are three sets of training data with test year from 2014 to 2016. We applied mean-scale transformation and deseasonalisation on the time series. According to the numerical experiment, deseasonalisation indeed accelerates the convergence of RNN.

We use three grouping schemes to group the food products.

Grouping by sales volume and volatility. The first grouping scheme is based on the same criteria as pharma product segmentation, which is using the volume and volatility of product sales/demands. We retain the drug grouping convention of dividing the time series into four groups: **HL**, **HH**, **LL**, and **LH**. The statistics regarding product sales of each group are summarized in Table 5.5

Table 5.5: Sales in four groups based on volume/volatility

Group Name	Mean	Median	CV.	Min	Max	# of products	# of Obser.
HL	17.34	11.00	1.36	0.00	247.00	68	18,632
HH	13.04	7.00	1.49	0.00	230.00	40	10,960
LL	3.62	3.00	1.12	0.00	33.00	40	10,960
LH	3.34	2.00	1.27	0.00	95.00	68	18,632

Due to the lack of relevant domain knowledge, the remaining two schemes are based on time series clustering.

Clustering by statistical features. The first clustering metric is the statistical features that describe the dynamics of time series, including mean, standard deviation, the strength

of linearity, the strength seasonality, the strength of trend, and the first order of autocorrelation [55]. We map each time series to a feature vector containing statistical feature values and use the Euler distance between the two time series feature vectors to evaluate their similarity. The clustering algorithm we applied for statistical feature vectors is K-Means. As described in Section 4.1.5, we cluster the sales series into three groups by exploring the elbow point of the Davies-Bouldin index (DBI) curve. Table 5.6 summarizes the statistics of product sales in each cluster.

Table 5.6: Sales in three clusters based on statistical features

Group Name	Mean	Median	CV.	Min	Max	# of products	# of Obser.
STAT_1	5.24	4.00	1.20	0.00	95.00	174	47,676
STAT_2	56.06	55.00	0.74	0.00	247.00	10	2,740
STAT_3	18.75	16.00	1.01	0.00	192.00	32	8,768

Clustering by time series distance. The second clustering metric is dynamic time warping (DTW) distance, which is used to evaluate the shape similarity of two time series [95]. We use DBSCAN and the DTW distance matrix to cluster the time series. By balancing the clustering quality (i.e., DBI) and sample size in each clusters, we get two groups of time series. The statistics of product sales in each cluster are summarized in Table 5.7.

Table 5.7: Sales in two clusters based on dynamic time warping distance

Group Name	Mean	Median	CV.	Min	Max	# of products	# of Obser.
DTW_1	3.44	2.00	1.21	0.00	95.00	118	32,332
DTW_2	15.75	10.00	1.41	0.00	247.00	98	26,852

The retail dataset also contains the product weekly price which is now served as an additional feature. Market observations and research literature reported that price fluctuations may affect the sales of goods [124, 104].

On the retail sales dataset, we conducted experiments similar to drug demand prediction and used the same naming convention: (model name)_(grouping scheme)_(additional feature). Recall that **STAT** represents a grouping scheme based on statistical features, and **prc** indicates that the model uses price as an additional feature. For example, **LR_STAT** is a linear regression model with the grouping scheme based on statistical

feature while **RNN_DTW_prc** is a RNN with the grouping scheme based on clustering and uses price as an additional feature.

• *Benefits of cross-products learning.*

In the numerical experiments on the retail sales dataset, we also use moving average (MA), exponential smoothing (ES) and basic linear regression (LR) trained on single time series as our baseline models. Advanced models including support vector regression (SVR), random forest (RF), fully Connected neural network (FC) and recurrent neural network (RNN) are optimized by cross-series training. Cross-series linear regression is used as a comparison reference. As mentioned, Vector Autoregression (VAR) also utilizes cross-series training, but instead of making the time series share one temporal variable, it builds an independent temporal variable for each time series. Table 5.8 compares the performance of our cross-series learning models with VAR and the baseline models.

Table 5.8: Forecasting bias and accuracy measures of cross training models using all food products

	Baselines			With Cross-product Training					
	MA	ES	LR	VAR	LR	SVR	RF	FC	RNN
NMSE	0.76 ± 0.08	0.83 ± 0.03	0.73 ± 0.07	0.95 ± 0.14	0.55 ± 0.03	0.60 ± 0.03	0.54 ± 0.04	0.55 ± 0.03	0.32 ± 0.10
NMAE	0.47 ± 0.01	0.47 ± 0.02	0.43 ± 0.02	0.55 ± 0.07	0.40 ± 0.01	0.40 ± 0.01	0.40 ± 0.01	0.40 ± 0.01	0.35 ± 0.03
Bias	0.00 ± 0.03	0.01 ± 0.03	-0.03 ± 0.04	0.13 ± 0.02	0.00 ± 0.02	-0.07 ± 0.02	0.03 ± 0.01	-0.01 ± 0.02	0.06 ± 0.00

VAR also does not apply to retail dataset settings. Before applying cross-series learning, we observe that the performance of advanced models (such as SVR, RF, and neural networks) is comparable to or even worse than basic linear regression without cross-series learning. This situation can be caused by severe overfitting problems from using more advanced models. From Table 5.8, we can see that apart from VAR, all cross-series models have achieved significant improvements, especially RNN. Therefore, for retail sales forecasting problems, data limitations remain to be a bottleneck for adopting the more advanced models to help enhance sales forecasting performance.

• *Global training v.s. group training*

From the extensive numerical experiments on the two pharma datasets, we have no-

ticed that if we directly train the model on the global time series, the prediction results of many low volume and high volatility drugs will be inaccurate. One possible reason is that the training process tends to satisfy samples that have a large impact on the loss function (i.e., high volume products). However, even if the time series are normalized in advance, the performance of the cross-series model on low volume products still has no improvement. Even worse, normalization makes the training process lose focus on high volume products, resulting in a decrease in the accuracy of high volume products. We believe this is caused by products with different volumes and volatility following different behavior patterns. Hence, the grouping scheme based on volume and volatility is proposed. To facilitate discussion, we now refer to the model trained on the global time series as the **global training model**, and the model trained on the subgroups as the **group training model**. In the retail sales forecasting, we also compare the performance of the global training model and the group training model based on volume and volatility in Table 5.9.

Table 5.9: Improvement of cross-products training models with grouping by volume/volatility

		Best Baseline	LR	LR_4	RNN	RNN_4
NMSE	All Products	0.73	24.74%	24.75%	56.47%	55.39%
	HL	0.35	7.61%	9.57%	47.67%	43.43%
	HH	0.90	40.15%	40.58%	69.60%	66.02%
	LL	1.10	53.95%	52.95%	59.64%	71.86%
	LH	1.84	57.16%	55.11%	60.81%	79.24%
NMAE	All Products	0.43	6.57%	7.30%	17.60%	25.60%
	HL	0.34	0.28%	0.00%	16.17%	17.37%
	HH	0.46	6.34%	4.60%	24.87%	21.00%
	LL	0.57	5.46%	10.39%	10.71%	37.13%
	LH	0.72	11.68%	15.50%	14.24%	45.69%

Table 5.9 indicates that global training is beneficial to both high volume and low volume products. At the same time, the RNN model trained by the group has further improved the NMSE of **LL** product sales forecasts from 59.64% to 71.68% and the NMSE of **LH** product sales forecasts from 60.81% to 79.24%. Therefore, in the retail sales forecasting, although low volume foods are not affected by global training like low volume drugs, they still have the potential for further improvement.

Group training optimize the sample quality to make the training process focus on the temporal patterns shared by similar time series. For general purpose predictions without domain knowledge, time series can also be grouped by clustering. In this case, the choice of criteria for time series similarity will affect the final performance as will be seen in the following discussion.

The first similarity considered is the dynamic characteristics of the time series. We use a statistical feature vector to describe the dynamics of time series. Table 5.10 lists the performance of the global training model and group training model for all products and each cluster. By comparing the performance of the global training RNN model and the group training RNN model on clusters, it can be seen that the **STAT_1** with the largest sample size benefits the most from the group training, and the accuracy of **STAT_2** deteriorates due to the small sample size. This observation also exposes the problem of clustering-based grouping, that is, uneven sample size distribution. Clustering can generate subgroups with high similarity, but it will also lead to an insufficient sample size of individual groups.

Table 5.10: Improvement of cross-products training models with clustering by statistical features

		Best Baseline	LR	LR_STAT	RNN	RNN_STAT
NMSE	All Products	0.73	24.74%	20.30%	56.47%	53.84%
	STAT_1	0.81	33.97%	34.25%	45.88%	63.26%
	STAT_2	0.20	19.10%	19.80%	67.30%	45.68%
	STAT 3	0.47	22.28%	20.41%	57.23%	63.76%
NMAE	All Products	0.43	6.57%	7.70%	17.60%	28.94%
	STAT_1	0.52	8.42%	10.11%	9.47%	35.38%
	STAT_2	0.30	5.15%	5.49%	35.74%	12.27%
	STAT_3	0.40	3.78%	4.32%	21.99%	31.43%

Another clustering criterion is the similarity of time series shapes which is discussed in Section 4.1. In this method, we balanced the clustering quality and sample size. Table 5.11 presents the performance of the global training model and group training model for all products and the two clusters(**DTW_1** and **DTW_2**).

Table 5.11: Improvement of cross-products training models with clustering by DTW

		Best Baseline	LR	LR_DTW	RNN	RNN_DTW
NMSE	All Products	0.73	24.74%	20.52%	56.47%	62.49%
	DTW_1	1.54	51.09%	50.00%	61.15%	77.76%
	DTW_2	0.49	19.56%	19.53%	59.64%	61.78%
NMAE	All Products	0.43	6.57%	7.07%	17.60%	22.70%
	DTW_1	0.66	14.08%	16.17%	13.50%	38.23%
	DTW_2	0.38	3.55%	3.42%	21.58%	16.47%

By using DTW based clustering, we obtain a group of time series (i.e., **DTW_1**) on which the baseline model has poor performance. Compared with the globally training RNN, the DTW based group training RNN significantly improves the NMSE on **DTW_1** from 61.15% to 77.76%. In addition to equal-length time series, DTW distance can also measure the similarity between the time series with different lengths or asynchronous sampling rates. Another advantage of using the shape-based similarity is its ability to compare various types of time series (e.g., physical trajectories).

The benefits of different grouping schemes for linear regression and RNN are shown in Table 5.12. Compared with linear regression, the benefits of the cross-series RNN from the grouping scheme are more significant. Group training RNN models have achieved consistent improvements for low volume products on all metrics.

Table 5.12: Improvement of cross-products training models with grouping schemes

		Best Baseline	LR	LR_4	LR_STAT	LR_DTW	RNN	RNN_4	RNN_STAT	RNN_DTW
NMSE	All Products	0.73	24.74%	24.75%	20.30%	20.52%	56.47%	55.39%	53.84%	62.49%
	HL	0.35	7.61%	9.57%	4.78%	5.03%	47.67%	43.43%	36.73%	53.27%
	HH	0.90	40.15%	40.58%	36.56%	37.94%	69.60%	66.02%	72.28%	72.41%
	LL	1.10	53.95%	52.95%	49.10%	48.01%	59.64%	71.86%	69.06%	73.67%
	LH	1.84	57.16%	55.11%	51.48%	50.23%	60.81%	79.24%	74.02%	79.35%
NMAE	All Products	0.43	6.57%	7.30%	7.70%	7.07%	17.60%	25.60%	28.94%	22.70%
	HL	0.34	0.28%	0.00%	2.06%	2.06%	16.17%	17.37%	22.67%	14.16%
	HH	0.46	6.34%	4.60%	8.47%	8.47%	24.87%	21.00%	37.65%	20.30%
	LL	0.57	5.46%	10.39%	12.67%	12.67%	10.71%	37.13%	30.68%	34.76%
	LH	0.72	11.68%	15.50%	17.40%	17.40%	14.24%	45.69%	36.90%	40.07%

According to the above experiments, product segmentation information (i.e., volume and volatility) is beneficial for time series grouping. In the absence of domain knowledge,

clustering-based methods can also achieve comparable improvements but need to take into account the number of samples in the group.

• *Sales forecasting with price information*

The results of using downstream inventory information for pharma demand forecasting indicate that features with strong contemporaneous impact can help improve the accuracy of the forecast. For retail products, sales may be correlated with the selling price. Therefore, we introduce price information as an additional feature. Table 5.12 shows RNN's performance for models using and not using price information for various grouping schemes.

Table 5.13: Improvement of cross-products training models by using price information

		Best Baseline	RNN	RNN_prc	RNN_4	RNN_4_prc	RNN_STAT	RNN_STAT_prc	RNN_DTW	RNN_DTW_prc
NMSE	All Products	0.73	56.47%	55.75%	55.39%	58.35%	53.84%	56.07%	62.49%	61.44%
	HL	0.35	47.67%	42.92%	43.43%	43.29%	36.73%	40.61%	53.27%	50.29%
	HH	0.90	69.60%	72.71%	66.02%	72.18%	72.28%	72.53%	72.41%	71.71%
	LL	1.10	59.64%	58.81%	71.86%	76.91%	69.06%	69.29%	73.67%	78.73%
	LH	1.84	60.81%	61.69%	79.24%	81.92%	74.02%	74.74%	79.35%	80.99%
NMAE	All Products	0.43	17.60%	23.58%	25.60%	28.17%	28.94%	30.83%	22.70%	23.67%
	HL	0.34	16.17%	16.64%	17.37%	17.64%	22.67%	24.90%	14.16%	12.96%
	HH	0.46	24.87%	25.61%	21.00%	24.44%	37.65%	38.09%	20.30%	21.24%
	LL	0.57	10.71%	10.84%	37.13%	41.87%	30.68%	28.58%	34.76%	41.38%
	LH	0.72	14.24%	14.36%	45.69%	51.33%	36.90%	35.73%	40.07%	43.28%

The price information helps improve the forecasting accuracy of RNN models on low volume products when grouping by volume & volatility and DTW based clustering. Moreover, both the accuracy of **RNN_STAT_prc** on high volume low volatility products and the accuracy of **RNN_4_prc** on high volume high volatility products benefit from price information across all metrics. In summary, using price information as an additional feature achieves improvements for certain products.

5.3 Conclusion

In this chapter, we show the generalizability of cross-series learning and introduce our generic cross-series learning framework. The framework includes all operations required for cross-series learning into four components, which are: preprocessing, grouping, cross-

series training, and postprocessing. To validate the performance of the forecasting framework, we tested the cross-series learning models for pharma demand forecasting on the second pharma demand dataset. The experimental results are consistent with the insights obtained from the first pharma dataset. Afterward, we conducted extensive experiments on a retail dataset using our generic cross-series learning framework. Based on the cross-series model's performance, we confirm the effectiveness of the generic framework and obtain the following observations:

- Cross-series learning can overcome the data limitation of high dimensional time series and enhance the performance of advanced models.
- Grouping by product volume and volatility is beneficial to sales and demand forecasting.
- Clustering can generate groups with similar time series but the sample size is unevenly distributed. The balance between group quality and the sample size is necessary for grouping based on time series clustering.
- In the absence of domain knowledge, grouping by time series clustering can achieve comparable improvements.
- The accuracy of group training RNN models on certain subgroups benefits from the price information.

In summary, cross-series learning and grouping schemes are generalizable to the forecasting problems in other fields.

Chapter 6

Conclusion

High dimensional time series with a large number of zeros often appear in supply chain demand, retail sales, etc. In industry, accurate and reliable prediction of high dimensional time series is critical. Manufacturers need demand forecasts for supply chain planning; power plants need energy consumption forecasts for resource allocation; retailers need sales forecasts for business management, among others. However, sparse high dimensional time series prediction faces three problems in practical applications: (1) simple models fail to capture complex patterns, (2) insufficient data prevents us from pursuing more advanced models, and (3) time series in the same dataset may have widely different properties. These problems prevent the current prevalent methods in the industry from providing reliable predictions, and theoretically successful advanced models fail to work in actual use.

To overcome the challenges in the sparse high dimensional time series forecasting, we started with a pharma demand forecasting problem, which predicts drug products future demand by using their historical order quantity. We developed a cross-series learning framework that trains a machine learning model on multiple related time series and uses cross-series information to improve prediction accuracy. Cross-series learning allows us to explore more advanced models, including support vector regression, random forests, and neural networks. We adopted three grouping schemes based on domain knowledge

and time series clustering to balance the tradeoff between sample size and sample quality. We introduced the downstream inventory information as an additional feature to assist demand forecasting. Compared with the benchmark models, our cross-series learning models achieved significant improvements in pharma demand forecasting.

To verify the generalizability of cross-series learning, we tested the pharma demand forecasting framework on another pharma demand dataset and obtained the observations consistent with the insights from the first dataset. We developed a generic framework that contains the operations required for cross-series learning from data preprocessing, grouping, cross-series training to postprocessing. Extensive experiments were conducted on a retail dataset to validate the effectiveness of the generic cross-series learning framework. We further confirmed the benefits of cross-series learning for advanced models, especially RNN. In addition to grouping schemes based on product characteristics, we also use time series clustering to group time series without domain knowledge. The criterion of clustering is the time series dynamic features and shape similarity. Finally, the price information is introduced to support retails sales forecasting.

Experimental results provide us following insights:

- Cross-series learning overcomes the data limitations of high dimensional time series and improves the prediction accuracy of advanced machine learning models.
- For product demand and sales forecasting, low volume products benefit the most from cross-series learning, because the low volume products have more severe data shortages than the high volume products.
- Cross-series training with different grouping schemes based on product-specific information, either by demand volume/volatility or by product-based domain knowledge (ATC code in our case), is effective. In the absence of domain knowledge, grouping by time series clustering can also achieve comparable improvements, but the balance between cluster quality and the sample size is required.

- Due to the unique feedback architecture, cross-series RNNs are always superior to other machine learning models. Moreover, RNN can capture intricate non-linear patterns, such as spikes in pharma demand. At the same time, cross-series learning provides RNN with sufficient data to construct a deeper network with more neurons.
- Introducing features with a strong contemporaneous impact is indeed helpful for enhancing the prediction accuracy, for example, pharma demand forecasting benefits from downstream inventory.

In this dissertation, we combine optimized cross-series learning technology with advanced machine learning models to generate accurate and reliable predictions for sparse high dimensional time series. Our cross-series learning framework can be applied to pharma manufacturers, wholesalers, and possibly other industries based on its robust performances. The dissertation provides practical guidelines for executing such a framework with corresponding domain knowledge. At the same time, our experimental results can provide a reference for other academic researches related to sparse high-dimensional time series.

Bibliography

[1] Abubakar Abid and James Zou. Autowarp: learning a warping distance from unlabeled time series using sequence autoencoders. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10568–10578. Curran Associates Inc., 2018.

[2] Y. Aviv. On the benefits of collaborative forecasting partnerships between retailers and manufacturers. *Management Science*, 53(5):777–794, 2007.

[3] L. Baardman, I. Levin, G. Perakis, and D. Singhvi. Leveraging comparables for new product sales forecasting. *Production and Operations Management*, 27(12):2339–2349, 2018.

[4] Kasun Bandara, Christoph Bergmeir, and Hansika Hewamalage. Lstm-msnet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[5] Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, 140:112896, 2020.

[6] Kasun Bandara, Peibei Shi, Christoph Bergmeir, Hansika Hewamalage, Quoc Tran, and Brian Seaman. Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In *International Conference on Neural Information Processing*, pages 462–474. Springer, 2019.

[7] Sumanta Basu, George Michailidis, et al. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4):1535–1567, 2015.

[8] C. Bergmeir, R.J Hyndman, and B. Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83, 2018.

[9] D.J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of AAAI Workshop on Knowledge Discovery in Databases, 1994*, pages 359–370, 1994.

[10] Philippe C Besse, Brendan Guillouet, Jean-Michel Loubes, and François Royer. Review and perspective for distance-based clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 17(11):3306–3317, 2016.

[11] T. Boone, R. Ganeshan, R.L Hicks, and N.R Sanders. Can google trends improve your sales forecast? *Production and Operations Management*, 2018.

[12] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[13] George EP Box and David A Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526, 1970.

[14] Melek Acar Boyacioglu and Derya Avci. An adaptive network-based fuzzy inference system (anfis) for the prediction of stock market return: the case of the istanbul stock exchange. *Expert Systems with Applications*, 37(12):7908–7912, 2010.

[15] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[16] Peter J Brockwell and Richard A Davis. *Introduction to time series and forecasting*. springer, 2016.

[17] Robert Goodell Brown. *Statistical forecasting for inventory control*. McGraw/Hill, 1959.

[18] G.P Cachon and M. Fisher. Supply chain inventory management and the value of shared information. *Management science*, 46(8):1032–1048, 2000.

[19] G.P Cachon and M. Olivares. Drivers of finished-goods inventory in the U.S. automobile industry. *Management science*, 56(1):202–216, 2010.

[20] R. Carbonneau, K. Laframboise, and R. Vahidov. Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, 184(3):1140–1154, 2008.

[21] Kanad Chakraborty, Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. Forecasting the behavior of multivariate time series using neural networks. *Neural networks*, 5(6):961–970, 1992.

[22] Nicolas Chapados. Effective bayesian modeling of groups of related count time series. *arXiv preprint arXiv:1405.3738*, 2014.

[23] C.W Chase. *Demand-driven forecasting: a structured approach to forecasting*. John Wiley & Sons, 2013.

[24] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[25] Lei Chen and Raymond Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803, 2004.

[26] Lei Chen, M Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502, 2005.

[27] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[28] T.M. Choi, S.W. Wallace, and Y. Wang. Big data analytics in operations management. *Production and Operations Management*, 27(10):1868–1883, 2018.

[29] Robert B Cleveland et al. Stl: A seasonal-trend decomposition procedure based on loess. 1990. *DOI: citeulike-article-id*, 1435502, 1990.

[30] A.G Cook. *Forecasting for the pharmaceutical industry: models for new product and in-market forecasting and how to use them*. Gower, 2016.

[31] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[32] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473, 2006.

[33] Grzegorz Dudek. Short-term load forecasting using random forests. In *Intelligent Systems' 2014*, pages 821–828. Springer, 2015.

[34] Jianqing Fan, Jinchi Lv, and Lei Qi. Sparse high-dimensional models in economics. *Annu. Rev. Econ.*, 3(1):291–317, 2011.

[35] K.J. Ferreira, B.H. Lee, and D. Simchi-Levi. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 18(1):69–88, 2015.

[36] M5 forecasting accuracy. https://www.kaggle.com/c/m5-forecasting-accuracy, 2020.

[37] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[38] Rui Fu, Zuo Zhang, and Li Li. Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 324–328. IEEE, 2016.

[39] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3):183–192, 1989.

[40] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

[41] C. García-Ascanio and C. Maté. Electric power demand forecasting using interval time series: A comparison between VAR and iMLP. *Energy Policy*, 38(2):715–725, 2010.

[42] Everette S Gardner Jr. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):1–28, 1985.

[43] John Clifford Gower. Properties of euclidean and non-euclidean distance matrices. *Linear Algebra and its Applications*, 67:81–97, 1985.

[44] Y. Grushka-Cockayne, Victor R.R Jose, and Lichtendahl J.K.C. Ensembles of over-fit and overconfident forecasts. *Management Science*, 63(4):1110–1130, 2016.

[45] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, NJ, 1994.

[46] Jun Han and Qiang Liu. Bootstrap model aggregation for distributed statistical learning. In *Advances in Neural Information Processing Systems*, pages 1795–1803, 2016.

[47] T. Hill, M. O'Connor, and W. Remus. Neural network models for time series forecasts. *Management science*, 42(7):1082–1092, 1996.

[48] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[49] Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.

[50] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[51] R.J Hyndman, R.A Ahmed, G. Athanasopoulos, and H.L Shang. Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*, 55(9):2579–2589, 2011.

[52] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[53] R.J. Hyndman and A.B. Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[54] Rob J Hyndman, Anne B Koehler, Ralph D Snyder, and Simone Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of forecasting*, 18(3):439–454, 2002.

[55] Rob J Hyndman, Earo Wang, and Nikolay Laptev. Large-scale unusual time series detection. In *2015 IEEE international conference on data mining workshop (ICDMW)*, pages 1616–1619. IEEE, 2015.

[56] C.L. Jain. Benchmarking forecasting practices in pharmaceutical industry. In *Proceedings of Pharmaceutical SAS Users Group*, 2003.

[57] Gardner J.E.S. Evaluating forecast performance in an inventory control system. *Management Science*, pages 490–499, 1990.

[58] Spencer S Jones, R Scott Evans, Todd L Allen, Alun Thomas, Peter J Haug, Shari J Welch, and Gregory L Snow. A multivariate time series approach to modeling and forecasting demand in the emergency department. *Journal of biomedical informatics*, 42(1):123–139, 2009.

[59] Katarina Juselius. *The cointegrated VAR model: methodology and applications*. Oxford university press, 2006.

[60] A. Karpathy. The unreasonable effectiveness of recurrent neural networks. *Andrej Karpathy blog*, 2015.

[61] Joe K Kearney and Stuart Hansen. Stream editing for animation. Technical report, IOWA UNIV IOWA CITY DEPT OF COMPUTER SCIENCE, 1990.

[62] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE, 2014.

[63] Mehdi Khashei and Mehdi Bijari. An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with applications*, 37(1):479–489, 2010.

[64] D. Kiely. The state of pharmaceutical industry supply planning and demand forecasting. *The Journal of Business Forecasting*, 23(3):20, 2004.

[65] Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319, 2003.

[66] W. Kim, J.H. Won, S. Park, and J. Kang. Demand forecasting models for medicines through wireless sensor networks data and topic trend analysis. *International Journal of Distributed Sensor Networks*, 11(9):907169, 2015.

[67] M. Kremer, E. Siemsen, and D.J Thomas. The sum and its parts: Judgmental hierarchical forecasting. *Management Science*, 62(9):2745–2764, 2015.

[68] Manish Kumar and M Thenmozhi. Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper*, 2006.

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

[70] Clifford Lam, Qiwei Yao, et al. Factor modeling for high-dimensional time series: inference for the number of factors. *The Annals of Statistics*, 40(2):694–726, 2012.

[71] Martin Längkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

[72] R.Y.K. Lau, W. Zhang, and W. Xu. Parallel aspect-oriented sentiment analysis for sales forecasting with big data. *Production and Operations Management*, 27(10):1775–1794, 2018.

[73] Jiaokun Liu, Erjia Cui, Haoqiang Hu, Xiaowei Chen, Xiqun Michael Chen, and Feng Chen. Short-term forecasting of emerging on-demand ride services. In *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, pages 489–495. IEEE, 2017.

[74] X. Liu, P.V. Singh, and K. Srinivasan. A structured analysis of unstructured big data by leveraging cloud computing. *Marketing Science*, 35(3):363–388, 2016.

[75] Chi-Jie Lu, Tian-Shyug Lee, and Chih-Chou Chiu. Financial time series forecasting using independent component analysis and support vector regression. *Decision support systems*, 47(2):115–125, 2009.

[76] S. Makridakis, S.C. Wheelwright, and R.J Hyndman. *Forecasting methods and applications*. John wiley & sons, 2008.

[77] Spyros Makridakis, Allan Andersen, Robert Carbone, Robert Fildes, Michele Hibon, Rudolf Lewandowski, Joseph Newton, Emanuel Parzen, and Robert Winkler. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of forecasting*, 1(2):111–153, 1982.

[78] Spyros Makridakis and Michele Hibon. The m3-competition: results, conclusions and implications. *International journal of forecasting*, 16(4):451–476, 2000.

[79] M Marseguerra, S Minoggio, A Rossi, and Enrico Zio. Neural networks prediction and fault diagnosis applied to stationary and non stationary arma modeled time series. *Progress in Nuclear Energy*, 27(1):25–36, 1992.

[80] Marion Maturilli, Andreas Herber, and Gert König-Langlo. Climatology and time series of surface meteorology in ny¬ ålesund, svalbard. *Earth System Science Data*, 5:155–163, 2013.

[81] G. Merkuryeva, A. Valberga, and A. Smirnov. Demand forecasting in pharmaceutical supply chains: A case study. *Procedia Computer Science*, 149:3–10, 2019.

[82] George Michailidis and Florence d'Alché Buc. Autoregressive models for gene regulatory network inference: Sparsity, stability and causality issues. *Mathematical biosciences*, 246(2):326–334, 2013.

[83] K-R Muller, Sebastian Mika, Gunnar Ratsch, Koji Tsuda, and Bernhard Scholkopf. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, 12(2):181–201, 2001.

[84] K-R Müller, Alexander J Smola, Gunnar Rätsch, Bernhard Schölkopf, Jens Kohlmorgen, and Vladimir Vapnik. Predicting time series with support vector ma-

chines. In *International Conference on Artificial Neural Networks*, pages 999–1004. Springer, 1997.

[85] Michael Nelson, Tim Hill, William Remus, and Marcus O'Connor. Time series forecasting using neural networks: Should the data be deseasonalized first? *Journal of forecasting*, 18(5):359–367, 1999.

[86] Paul Newbold. Arima model building and the time series analysis approach to forecasting. *Journal of Forecasting*, 2(1):23–35, 1983.

[87] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[88] World Health Organization. *Introduction to drug utilization research*. World Health Organization, 2003.

[89] B. Pan, D. Chenguang Wu, and H. Song. Forecasting hotel room demand using search engine data. *Journal of Hospitality and Tourism Technology*, 3(3):196–210, 2012.

[90] C Carl Pegels. Exponential forecasting: some new variations. *Management Science*, pages 311–315, 1969.

[91] Edie M Rasmussen. Clustering algorithms. *Information retrieval: data structures & algorithms*, 419:442, 1992.

[92] V. Richard. Demand shaping: Achieving and maintaining optimal supply-and-demand alignment. *SAS - White paper*, pages 1–12, 2014.

[93] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[94] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[95] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[96] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 2019.

[97] Nicholas I Sapankevych and Ravi Sankar. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine*, 4(2):24–38, 2009.

[98] T. Sastri. A state space modeling approach for time series forecasting. *Management Science*, 31(11):1451–1470, 1985.

[99] Jürgen Schmidhuber. Habilitation thesis: System modeling and optimization. *Page 150 ff demonstrates credit assignment across the equivalent of 1,200 layers in an unfolded RNN*, 1993.

[100] D.C Schmittlein, J. Kim, and D.G Morrison. Combining forecasts: Operational adjustments to theoretically optimal rules. *Management Science*, 36(9):1044–1056, 1990.

[101] L.B Schwarz and H. Zhao. The unexpected impact of information sharing on us pharmaceutical supply chains. *Interfaces*, 41(4):354–364, 2011.

[102] E.W.K. See-To and E.W.T. Ngai. Customer reviews for demand distribution and sales nowcasting: a big data approach. *Annals of Operations Research*, 270(1-2):415–431, 2018.

[103] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE, 2017.

[104] Rashmi Sharma and Ashok K Sinha. Sales forecast of an automobile industry. *International Journal of Computer Applications*, 53(12), 2012.

[105] C.A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, pages 1–48, 1980.

[106] Stephen M Smith. The future of fmri connectivity. *Neuroimage*, 62(2):1257–1266, 2012.

[107] Alex J Smola and Peter L Bartlett. Sparse greedy gaussian process regression. In *Advances in neural information processing systems*, pages 619–625, 2001.

[108] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[109] H. Song and G. Li. Tourism demand modelling and forecasting—a review of recent research. *Tourism management*, 29(2):203–220, 2008.

[110] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4597–4605, 2015.

[111] Zaiyong Tang, Chrys De Almeida, and Paul A Fishwick. Time series forecasting using neural networks vs. box-jenkins methodology. *Simulation*, 57(5):303–310, 1991.

[112] Leonard J Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review. *International journal of forecasting*, 16(4):437–450, 2000.

[113] Hiro Y Toda and Peter CB Phillips. Vector autoregression and causality: a theoretical overview and simulation study. *Econometric reviews*, 13(2):259–285, 1994.

[114] Juan R Trapero, Nikolaos Kourentzes, and Robert Fildes. On the identification of sales forecasting models in the presence of promotions. *Journal of the operational Research Society*, 66(2):299–307, 2015.

[115] Vladimir Vovk. Kernel ridge regression. In *Empirical inference*, pages 105–116. Springer, 2013.

[116] Matt Weller and Sven F Crone. Supply chain forecasting: Best practices & benchmarking study, 2012.

[117] Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

[118] L. Xu, V. Mani, and H. Zhao. Not a box of nuts and bolts: Are specialty distributors the right channel for rising specialty drugs. *Working paper, Penn State*, 2018.

[119] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in neural information processing systems*, pages 847–855, 2016.

[120] George Udny Yule. Vii. on a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 226(636-646):267–298, 1927.

[121] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[122] Guoqiang Zhang, B Eddy Patuwo, and Michael Y Hu. Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14(1):35–62, 1998.

[123] H. Zhao, C. Xiong, S. Gavirneni, and A. Fein. Fee-for-service contracts in pharmaceutical distribution supply chains: design, analysis, and management. *Manufacturing & Service Operations Management*, 14(4):685–699, 2012.

[124] Kui Zhao and Can Wang. Sales forecast in e-commerce using convolutional neural network. *arXiv preprint arXiv:1708.07946*, 2017.

Appendices

.1 Performance of Cross-drug Forecasting Models

Table 1: Forecasting bias and accuracy measures of cross-drug training models grouped by volume/volatility

		Best Baseline	LR	LR_4	SVR	SVR_4	RF	RF_4	FC	FC_4	RNN	RNN_4
NMSE	All Drugs	1.67 ± 0.10	1.58 ± 0.09	1.55 ± 0.10	1.83 ± 0.11	1.66 ± 0.12	1.60 ± 0.10	1.91 ± 0.15	1.50 ± 0.09	1.73 ± 0.10	0.98 ± 0.12	0.89 ± 0.19
	HL	1.05 ± 0.10	1.01 ± 0.10	1.01 ± 0.10	1.07 ± 0.12	1.08 ± 0.12	1.13 ± 0.11	1.26 ± 0.16	1.01 ± 0.10	1.03 ± 0.10	0.64 ± 0.12	0.58 ± 0.15
	HH	1.44 ± 0.80	1.20 ± 0.65	1.36 ± 0.66	1.21 ± 0.34	1.50 ± 0.42	1.03 ± 0.30	1.51 ± 0.38	1.20 ± 0.35	2.68 ± 0.74	0.80 ± 0.27	0.76 ± 0.30
	LL	2.69 ± 0.55	2.44 ± 0.59	2.42 ± 0.69	8.31 ± 3.96	2.41 ± 0.70	2.29 ± 0.71	2.61 ± 0.55	27.59 ± 8.35	4.76 ± 1.33	4.15 ± 1.12	1.90 ± 0.37
	LH	17.37 ± 0.26	18.02 ± 0.43	17.66 ± 0.45	21.90 ± 2.05	18.24 ± 0.54	14.90 ± 0.34	15.76 ± 0.89	33.68 ± 2.25	18.01 ± 1.79	20.88 ± 1.98	8.72 ± 0.49
NMAE	All Drugs	0.30 ± 0.03	0.30 ± 0.03	0.30 ± 0.04	0.32 ± 0.05	0.29 ± 0.05	0.29 ± 0.04	0.33 ± 0.05	0.29 ± 0.03	0.32 ± 0.03	0.25 ± 0.04	0.25 ± 0.05
	HL	0.29 ± 0.04	0.28 ± 0.04	0.28 ± 0.04	0.30 ± 0.06	0.30 ± 0.06	0.30 ± 0.05	0.30 ± 0.06	0.29 ± 0.04	0.28 ± 0.05	0.26 ± 0.05	0.24 ± 0.07
	HH	0.37 ± 0.22	0.33 ± 0.20	0.36 ± 0.22	0.36 ± 0.19	0.41 ± 0.23	0.29 ± 0.18	0.41 ± 0.22	0.35 ± 0.18	0.58 ± 0.30	0.27 ± 0.17	0.28 ± 0.19
	LL	0.69 ± 0.20	0.66 ± 0.21	0.65 ± 0.22	2.25 ± 1.27	0.68 ± 0.24	0.68 ± 0.26	0.63 ± 0.26	4.96 ± 1.27	1.41 ± 0.54	0.87 ± 0.27	0.56 ± 0.12
	LH	0.81 ± 0.22	0.87 ± 0.19	0.88 ± 0.22	2.15 ± 0.41	1.17 ± 0.23	0.80 ± 0.16	0.73 ± 0.17	4.37 ± 0.53	1.34 ± 0.18	1.28 ± 0.21	0.65 ± 0.14
bias	All Drugs	-0.03 ± 0.02	-0.04 ± 0.02	-0.03 ± 0.02	0.03 ± 0.06	0.02 ± 0.06	0.01 ± 0.04	0.01 ± 0.06	0.02 ± 0.02	0.00 ± 0.04	0.01 ± 0.04	0.00 ± 0.06
	HL	-0.03 ± 0.02	-0.02 ± 0.03	-0.02 ± 0.03	0.02 ± 0.07	0.01 ± 0.07	0.01 ± 0.05	0.01 ± 0.09	0.01 ± 0.03	0.02 ± 0.04	0.01 ± 0.05	0.00 ± 0.07
	HH	-0.10 ± 0.27	-0.04 ± 0.11	-0.02 ± 0.28	-0.10 ± 0.08	0.10 ± 0.07	-0.06 ± 0.05	-0.02 ± 0.05	-0.04 ± 0.05	-0.26 ± 0.32	-0.05 ± 0.03	-0.04 ± 0.04
	LL	-0.27 ± 0.19	-0.02 ± 0.03	-0.16 ± 0.11	2.02 ± 1.25	0.01 ± 0.17	0.15 ± 0.12	0.12 ± 0.22	4.89 ± 1.16	0.57 ± 0.70	0.17 ± 0.23	0.13 ± 0.14
	LH	-0.64 ± 0.09	-0.06 ± 0.04	-0.56 ± 0.16	1.49 ± 0.50	-0.13 ± 0.23	-0.14 ± 0.09	-0.07 ± 0.07	3.94 ± 0.52	-0.10 ± 0.32	0.50 ± 0.12	0.01 ± 0.08

Table 2: Forecasting bias and accuracy measures of cross-drug training models grouped by ATC codes

		Best Baseline	LR	LR_ATC	SVR	SVR_ATC	RF	RF_ATC	FC	FC_ATC	RNN	RNN_ATC
NMSE	All Drugs	1.67	5.4%	6.0%	-9.6%	-1.2%	4.2%	-18.0%	10.2%	-1.2%	41.3%	47.3%
	A	1.46	6.2%	4.8%	5.5%	-8.9%	12.3%	8.2%	7.5%	0.7%	2.7%	24.0%
	B	0.55	14.5%	12.7%	12.7%	10.9%	16.4%	14.5%	16.4%	10.9%	38.2%	52.7%
	C	1.24	-0.8%	-10.5%	-8.1%	-21.0%	5.6%	14.5%	-4.0%	-17.7%	-29.8%	4.0%
	G	1.45	-9.0%	-1.4%	-49.0%	-47.6%	--%	--%	-20.0%	-42.1%	35.9%	37.9%
	J	1.20	10.0%	8.3%	-17.5%	-0.8%	2.5%	-20.0%	-21.7%	-2.5%	-6.7%	10.8%
	N	7.53	15.1%	3.6%	13.5%	-8.0%	24.8%	19.1%	14.7%	-10.8%	39.2%	42.1%
NMAE	All Drugs	0.30	0.0%	3.3%	-6.7%	-6.7%	3.3%	-10.0%	3.3%	-6.7%	16.7%	16.7%
	A	0.39	5.1%	5.1%	2.6%	-7.7%	7.7%	5.1%	7.7%	2.6%	2.6%	5.1%
	B	0.29	10.3%	10.3%	6.9%	3.4%	6.9%	6.9%	10.3%	3.4%	17.2%	27.6%
	C	0.35	-2.9%	0.0%	-11.4%	-8.6%	5.7%	8.6%	-14.3%	-8.6%	-2.9%	0.0%
	G	0.39	-2.6%	-2.6%	-25.6%	-25.6%	--%	--%	-10.3%	-25.6%	12.8%	15.4%
	J	0.36	5.6%	2.8%	-55.6%	-8.3%	16.7%	-30.6%	-11.1%	-11.1%	-16.7%	13.9%
	N	0.42	9.5%	2.4%	9.5%	-11.9%	19.0%	11.9%	23.8%	-11.9%	19.0%	16.7%

Table 3: Forecasting bias and accuracy measures of cross-drug training models grouped by DTW

		Best Baseline	LR	LR_4	LR_ATC	LR_DTW	RNN	RNN_4	RNN_ATC	RNN_DTW
NMSE	All Drugs	1.67 ± 0.10	1.58 ± 0.09	1.55 ± 0.10	1.57 ± 0.10	1.54 ± 0.09	0.98 ± 0.12	0.89 ± 0.19	0.88 ± 0.16	0.78 ± 0.12
	HL	1.05 ± 0.10	1.01 ± 0.10	1.01 ± 0.10	1.02 ± 0.10	1.01 ± 0.10	0.64 ± 0.12	0.58 ± 0.15	0.58 ± 0.15	0.52 ± 0.13
	HH	1.44 ± 0.80	1.20 ± 0.65	1.36 ± 0.66	1.37 ± 0.42	1.25 ± 0.36	0.80 ± 0.27	0.76 ± 0.30	0.65 ± 0.25	0.61 ± 0.24
	LL	2.69 ± 0.55	2.44 ± 0.59	2.42 ± 0.69	2.71 ± 0.64	2.53 ± 0.60	4.15 ± 1.12	1.90 ± 0.37	4.71 ± 0.61	2.43 ± 0.60
	LH	17.37 ± 0.26	18.02 ± 0.43	17.66 ± 0.45	20.15 ± 0.55	17.68 ± 0.42	20.88 ± 1.98	8.72 ± 0.49	12.95 ± 0.42	12.66 ± 0.43
NMAE	All Drugs	0.30 ± 0.03	0.30 ± 0.03	0.30 ± 0.04	0.29 ± 0.03	0.29 ± 0.03	0.25 ± 0.04	0.25 ± 0.05	0.25 ± 0.05	0.24 ± 0.05
	HL	0.29 ± 0.04	0.28 ± 0.04	0.28 ± 0.04	0.28 ± 0.04	0.28 ± 0.04	0.26 ± 0.05	0.24 ± 0.07	0.24 ± 0.08	0.23 ± 0.06
	HH	0.37 ± 0.22	0.33 ± 0.20	0.36 ± 0.22	0.35 ± 0.21	0.34 ± 0.19	0.27 ± 0.17	0.28 ± 0.19	0.26 ± 0.17	0.27 ± 0.17
	LL	0.69 ± 0.20	0.66 ± 0.21	0.65 ± 0.22	0.70 ± 0.22	0.67 ± 0.21	0.87 ± 0.27	0.56 ± 0.12	0.78 ± 0.31	0.60 ± 0.20
	LH	0.81 ± 0.22	0.87 ± 0.19	0.88 ± 0.22	0.84 ± 0.20	0.81 ± 0.16	1.28 ± 0.21	0.65 ± 0.14	0.71 ± 0.20	0.72 ± 0.19
bias	All Drugs	-0.03 ± 0.02	-0.04 ± 0.02	-0.03 ± 0.02	-0.02 ± 0.02	-0.01 ± 0.02	0.01 ± 0.04	0.00 ± 0.06	0.02 ± 0.05	0.02 ± 0.04
	HL	-0.03 ± 0.02	-0.02 ± 0.03	-0.02 ± 0.03	-0.01 ± 0.02	-0.01 ± 0.02	0.01 ± 0.05	0.00 ± 0.07	0.02 ± 0.08	0.03 ± 0.06
	HH	-0.10 ± 0.27	-0.04 ± 0.11	-0.02 ± 0.28	-0.04 ± 0.04	-0.05 ± 0.03	-0.05 ± 0.03	-0.04 ± 0.04	-0.02 ± 0.04	-0.06 ± 0.04
	LL	-0.27 ± 0.19	-0.02 ± 0.03	-0.16 ± 0.11	-0.08 ± 0.11	-0.07 ± 0.05	0.17 ± 0.23	0.13 ± 0.14	0.22 ± 0.03	0.14 ± 0.03
	LH	-0.64 ± 0.09	-0.06 ± 0.04	-0.56 ± 0.16	-0.17 ± 0.10	-0.28 ± 0.13	0.50 ± 0.12	0.01 ± 0.08	0.46 ± 0.03	0.38 ± 0.03

Table 4: Benefit of inventory information

		Best Baseline	RNN	RNN_inv	RNN_4	RNN_4_inv	RNN_ATC	RNN_ATC_inv	RNN_DTW	RNN_DTW_inv
NMSE	All Drugs	1.67 ± 0.10	0.98 ± 0.12	0.90 ± 0.14	0.89 ± 0.19	0.85 ± 0.14	0.88 ± 0.16	0.81 ± 0.14	0.78 ± 0.12	0.74 ± 0.10
	HL	1.05 ± 0.10	0.64 ± 0.12	0.59 ± 0.15	0.58 ± 0.15	0.55 ± 0.15	0.58 ± 0.15	0.53 ± 0.13	0.52 ± 0.13	0.49 ± 0.09
	HH	1.44 ± 0.80	0.80 ± 0.27	0.76 ± 0.25	0.76 ± 0.30	0.75 ± 0.26	0.65 ± 0.25	0.63 ± 0.27	0.61 ± 0.24	0.57 ± 0.21
	LL	2.69 ± 0.55	4.15 ± 1.12	3.80 ± 1.09	1.90 ± 0.37	1.05 ± 0.50	4.71 ± 0.61	3.47 ± 0.86	2.43 ± 0.60	2.31 ± 0.92
	LH	17.37 ± 0.26	20.88 ± 1.98	19.86 ± 0.78	8.72 ± 0.49	8.43 ± 0.37	12.95 ± 0.42	10.92 ± 0.63	12.66 ± 0.43	21.27 ± 0.48
NMAE	All Drugs	0.30 ± 0.03	0.25 ± 0.04	0.25 ± 0.05	0.25 ± 0.05	0.24 ± 0.05	0.25 ± 0.05	0.24 ± 0.05	0.24 ± 0.05	0.24 ± 0.04
	HL	0.29 ± 0.04	0.26 ± 0.05	0.24 ± 0.06	0.24 ± 0.07	0.23 ± 0.06	0.24 ± 0.08	0.23 ± 0.06	0.23 ± 0.06	0.23 ± 0.05
	HH	0.37 ± 0.22	0.27 ± 0.17	0.30 ± 0.18	0.28 ± 0.19	0.27 ± 0.17	0.26 ± 0.17	0.25 ± 0.17	0.27 ± 0.17	0.27 ± 0.16
	LL	0.69 ± 0.20	0.87 ± 0.27	0.84 ± 0.28	0.56 ± 0.12	0.51 ± 0.19	0.78 ± 0.31	0.72 ± 0.25	0.60 ± 0.20	0.68 ± 0.26
	LH	0.81 ± 0.22	1.28 ± 0.21	1.05 ± 0.21	0.65 ± 0.14	0.62 ± 0.15	0.71 ± 0.20	0.44 ± 0.23	0.72 ± 0.19	1.32 ± 0.22
bias	All Drugs	-0.03 ± 0.02	0.01 ± 0.04	0.03 ± 0.04	0.00 ± 0.06	0.02 ± 0.04	0.02 ± 0.05	0.01 ± 0.04	0.02 ± 0.04	0.01 ± 0.04
	HL	-0.03 ± 0.02	0.01 ± 0.05	0.03 ± 0.05	0.00 ± 0.07	0.02 ± 0.05	0.02 ± 0.08	0.01 ± 0.05	0.03 ± 0.06	0.01 ± 0.05
	HH	-0.10 ± 0.27	-0.05 ± 0.03	-0.02 ± 0.03	-0.04 ± 0.04	-0.04 ± 0.03	-0.02 ± 0.04	-0.02 ± 0.03	-0.06 ± 0.04	-0.06 ± 0.04
	LL	-0.27 ± 0.19	0.17 ± 0.23	0.19 ± 0.40	0.13 ± 0.14	0.12 ± 0.11	0.22 ± 0.03	0.13 ± 0.18	0.14 ± 0.03	0.11 ± 0.16
	LH	-0.64 ± 0.09	0.50 ± 0.12	0.40 ± 0.17	0.01 ± 0.08	-0.02 ± 0.11	0.46 ± 0.03	0.45 ± 0.19	0.38 ± 0.03	0.06 ± 0.07

Note that the notation “LR_DC” means a linear regression model is built using data from all DCs, while “LR_DC_each” means a linear regression model is fitted for each DC’s data. Note that if we train the model using information across DCs within the same trade partner, the second part will be “TP”, instead of “DC”. If we have “_inv,” that means we

include inventory data in the training step.

Table 5: NMSE of cross-drug training models using supply chain structure

	All	HL	HH	LL	LH
LR	1.58 ± 0.09	1.01 ± 0.10	1.20 ± 0.65	2.44 ± 0.59	18.02 ± 0.43
LR_DC	1.59 ± 0.09	1.01 ± 0.10	1.14 ± 0.75	2.18 ± 0.59	17.69 ± 0.36
LR_TP	1.58 ± 0.09	1.02 ± 0.09	1.13 ± 0.64	2.21 ± 0.61	17.83 ± 0.41
LR_4	1.55 ± 0.10	1.01 ± 0.10	1.36 ± 0.66	2.42 ± 0.69	17.66 ± 0.45
LR_DC_4	1.56 ± 0.09	1.05 ± 0.10	1.37 ± 0.64	2.37 ± 0.61	18.03 ± 0.42
LR_TP_4	1.55 ± 0.10	1.03 ± 0.10	1.42 ± 0.60	2.56 ± 0.66	19.26 ± 0.78
LR_ATC	1.57 ± 0.10	1.02 ± 0.10	1.37 ± 0.42	2.71 ± 0.64	20.15 ± 0.55
LR_DC_ATC	1.56 ± 0.09	1.02 ± 0.09	1.25 ± 0.61	2.69 ± 0.60	17.33 ± 0.40
LR_TP_ATC	1.56 ± 0.09	1.01 ± 0.09	1.32 ± 0.58	2.48 ± 0.62	17.21 ± 0.51
LR_DTW	1.54 ± 0.09	1.01 ± 0.10	1.25 ± 0.36	2.53 ± 0.60	17.68 ± 0.42
LR_DC_DTW	1.57 ± 0.09	1.03 ± 0.09	1.19 ± 0.58	2.21 ± 0.54	18.32 ± 0.39
LR_TP_DTW	1.56 ± 0.09	1.01 ± 0.09	1.21 ± 0.56	2.52 ± 0.55	16.69 ± 0.49
LR_inv	1.59 ± 0.09	1.03 ± 0.09	1.18 ± 0.61	2.17 ± 0.55	17.88 ± 0.39
LR_DC_inv	1.55 ± 0.09	1.02 ± 0.09	1.23 ± 0.59	2.42 ± 0.56	18.21 ± 0.41
LR_TP_inv	1.56 ± 0.10	1.03 ± 0.10	1.25 ± 0.57	2.36 ± 0.57	19.00 ± 0.51

Table 6: NMAE of cross-drug training models using supply chain structure

	All	HL	HH	LL	LH
LR	0.30 ± 0.03	0.28 ± 0.04	0.33 ± 0.20	0.66 ± 0.21	0.87 ± 0.19
LR_DC	0.30 ± 0.04	0.28 ± 0.04	0.31 ± 0.25	0.66 ± 0.21	0.82 ± 0.21
LR_TP	0.30 ± 0.04	0.28 ± 0.04	0.34 ± 0.20	0.67 ± 0.20	0.85 ± 0.19
LR_4	0.30 ± 0.04	0.28 ± 0.04	0.36 ± 0.22	0.65 ± 0.22	0.88 ± 0.22
LR_DC_4	0.30 ± 0.04	0.28 ± 0.04	0.35 ± 0.19	0.67 ± 0.20	0.87 ± 0.20
LR_TP_4	0.30 ± 0.05	0.28 ± 0.05	0.31 ± 0.21	0.65 ± 0.24	0.85 ± 0.31
LR_ATC	0.29 ± 0.03	0.28 ± 0.04	0.35 ± 0.21	0.68 ± 0.22	0.84 ± 0.20
LR_DC_ATC	0.29 ± 0.04	0.28 ± 0.04	0.36 ± 0.19	0.67 ± 0.20	0.83 ± 0.20
LR_TP_ATC	0.29 ± 0.04	0.28 ± 0.04	0.33 ± 0.20	0.66 ± 0.22	0.87 ± 0.25
LR_DTW	0.29 ± 0.03	0.28 ± 0.04	0.34 ± 0.19	0.67 ± 0.21	0.81 ± 0.16
LR_DC_DTW	0.30 ± 0.04	0.29 ± 0.04	0.34 ± 0.18	0.68 ± 0.19	0.83 ± 0.20
LR_TP_DTW	0.30 ± 0.04	0.29 ± 0.04	0.35 ± 0.20	0.65 ± 0.20	0.88 ± 0.24
LR_inv	0.29 ± 0.04	0.28 ± 0.04	0.31 ± 0.20	0.65 ± 0.19	0.85 ± 0.20
LR_DC_inv	0.29 ± 0.04	0.28 ± 0.04	0.32 ± 0.20	0.67 ± 0.20	0.85 ± 0.23
LR_TP_inv	0.29 ± 0.04	0.28 ± 0.04	0.32 ± 0.20	0.67 ± 0.20	0.86 ± 0.23

Table 7: Forecasting bias of cross-drug training models using supply chain structure

	All	HL	HH	LL	LH
LR	-0.04 ± 0.02	-0.02 ± 0.03	-0.04 ± 0.11	-0.02 ± 0.03	-0.06 ± 0.04
LR_DC	-0.03 ± 0.03	-0.02 ± 0.03	-0.02 ± 0.13	-0.05 ± 0.07	-0.11 ± 0.09
LR_TP	-0.04 ± 0.04	-0.02 ± 0.04	-0.02 ± 0.13	-0.08 ± 0.06	-0.23 ± 0.09
LR_4	-0.03 ± 0.02	-0.02 ± 0.03	-0.02 ± 0.28	-0.16 ± 0.11	-0.56 ± 0.16
LR_DC_4	-0.04 ± 0.02	-0.02 ± 0.03	-0.03 ± 0.14	-0.14 ± 0.07	-0.42 ± 0.09
LR_TP_4	-0.01 ± 0.06	-0.02 ± 0.06	-0.04 ± 0.14	-0.16 ± 0.25	-0.37 ± 0.25
LR_ATC	-0.02 ± 0.02	-0.01 ± 0.02	-0.04 ± 0.04	-0.08 ± 0.11	-0.17 ± 0.10
LR_DC_ATC	-0.03 ± 0.04	-0.01 ± 0.04	-0.04 ± 0.13	-0.09 ± 0.06	-0.20 ± 0.09
LR_TP_ATC	-0.02 ± 0.05	-0.02 ± 0.05	-0.02 ± 0.15	-0.08 ± 0.11	-0.18 ± 0.13
LR_DTW	-0.01 ± 0.02	-0.01 ± 0.02	-0.05 ± 0.03	-0.07 ± 0.05	-0.28 ± 0.13
LR_DC_DTW	-0.04 ± 0.04	-0.01 ± 0.04	-0.03 ± 0.15	-0.10 ± 0.06	-0.15 ± 0.09
LR_TP_DTW	-0.03 ± 0.05	-0.01 ± 0.05	-0.04 ± 0.16	-0.08 ± 0.08	-0.21 ± 0.18
LR_inv	-0.04 ± 0.04	-0.01 ± 0.04	-0.03 ± 0.14	-0.11 ± 0.06	-0.32 ± 0.09
LR_DC_inv	-0.03 ± 0.05	-0.02 ± 0.05	-0.02 ± 0.13	-0.07 ± 0.07	-0.16 ± 0.12
LR_TP_inv	-0.03 ± 0.05	-0.01 ± 0.05	-0.05 ± 0.14	-0.09 ± 0.09	-0.51 ± 0.19

Table 8: NMSE of cross-drug training models with inventory information

	All Drugs	HL	HH	LL	LH
LR	1.58 ± 0.09	1.01 ± 0.10	1.20 ± 0.65	2.44 ± 0.59	18.02 ± 0.43
LR_inv	1.59 ± 0.09	1.03 ± 0.09	1.18 ± 0.63	2.17 ± 0.60	17.88 ± 0.41
LR_4	1.55 ± 0.10	1.01 ± 0.10	1.36 ± 0.66	2.42 ± 0.69	17.66 ± 0.45
LR_4_inv	1.56 ± 0.09	1.02 ± 0.10	1.35 ± 0.63	2.31 ± 0.68	18.13 ± 0.40
LR_ATC	1.57 ± 0.10	1.02 ± 0.10	1.37 ± 0.42	2.71 ± 0.64	20.15 ± 0.55
LR_ATC_inv	1.57 ± 0.10	1.03 ± 0.09	1.36 ± 0.41	3.12 ± 0.65	19.21 ± 0.53
LR_DTW	1.54 ± 0.09	1.01 ± 0.10	1.25 ± 0.36	2.53 ± 0.60	17.68 ± 0.42
LR_DTW_inv	1.53 ± 0.09	1.02 ± 0.09	1.05 ± 0.37	2.51 ± 0.58	16.69 ± 0.43
RF	1.60 ± 0.10	1.13 ± 0.11	1.03 ± 0.30	2.29 ± 0.71	14.90 ± 0.34
RF_inv	1.94 ± 0.09	3.21 ± 0.10	1.17 ± 0.33	4.26 ± 0.69	21.13 ± 0.36
RF_4	1.91 ± 0.15	1.26 ± 0.16	1.51 ± 0.38	2.61 ± 0.55	15.76 ± 0.89
RF_4_inv	1.71 ± 0.16	2.93 ± 0.16	2.72 ± 0.39	2.43 ± 0.52	13.21 ± 0.84
RF_ATC	2.01 ± 0.21	2.82 ± 0.22	2.68 ± 0.31	23.51 ± 0.74	34.31 ± 1.28
RF_ATC_inv	2.41 ± 0.20	2.33 ± 0.21	2.71 ± 0.33	21.68 ± 0.74	33.18 ± 1.26
SVR	1.83 ± 0.11	1.07 ± 0.12	1.21 ± 0.34	8.31 ± 3.69	21.90 ± 2.05
SVR_inv	2.25 ± 0.10	2.27 ± 0.11	2.84 ± 0.35	6.11 ± 3.24	25.56 ± 1.28
SVR_4	1.66 ± 0.12	1.08 ± 0.12	1.50 ± 0.42	2.41 ± 0.70	18.24 ± 0.54
SVR_4_inv	1.54 ± 0.12	1.31 ± 0.11	1.42 ± 0.37	3.21 ± 0.71	17.28 ± 0.55
SVR_ATC	1.52 ± 0.12	1.45 ± 0.11	1.47 ± 0.47	4.32 ± 1.43	17.34 ± 0.48
SVR_ATC_inv	1.54 ± 0.11	1.32 ± 0.12	1.35 ± 0.48	3.60 ± 1.24	18.01 ± 0.50
FC	1.50 ± 0.09	1.01 ± 0.10	1.20 ± 0.35	27.59 ± 8.35	33.68 ± 2.25
FC_inv	1.62 ± 0.10	1.14 ± 0.10	1.28 ± 0.36	11.21 ± 7.63	16.54 ± 2.13
FC_4	1.73 ± 0.10	1.03 ± 0.10	2.68 ± 0.74	4.76 ± 1.33	18.01 ± 1.79
FC_4_inv	1.63 ± 0.09	1.02 ± 0.10	1.17 ± 0.72	5.89 ± 1.32	19.91 ± 1.66
FC_ATC	1.61 ± 0.13	1.01 ± 0.11	1.08 ± 0.48	18.21 ± 1.38	19.23 ± 0.52
FC_ATC_inv	1.56 ± 0.13	1.02 ± 0.12	1.09 ± 0.47	10.13 ± 1.34	18.89 ± 0.49
RNN	0.98 ± 0.12	0.64 ± 0.12	0.80 ± 0.27	4.15 ± 1.12	20.88 ± 1.98
RNN_inv	0.90 ± 0.14	0.59 ± 0.15	0.76 ± 0.25	3.80 ± 1.09	19.86 ± 0.78
RNN_4	0.89 ± 0.19	0.58 ± 0.15	0.76 ± 0.30	1.90 ± 0.37	8.72 ± 0.49
RNN_4_inv	0.85 ± 0.14	0.55 ± 0.15	0.75 ± 0.26	1.05 ± 0.50	8.43 ± 0.37
RNN_ATC	0.88 ± 0.16	0.58 ± 0.15	0.65 ± 0.25	4.71 ± 0.61	12.95 ± 0.42
RNN_ATC_inv	0.81 ± 0.14	0.53 ± 0.13	0.63 ± 0.27	3.47 ± 0.86	10.92 ± 0.63
RNN_DTW	0.78 ± 0.12	0.52 ± 0.13	0.61 ± 0.24	2.43 ± 0.60	12.66 ± 0.43
RNN_DTW_inv	0.74 ± 0.10	0.49 ± 0.09	0.57 ± 0.21	2.31 ± 0.92	21.27 ± 0.48

Table 9: NMAE of cross-drug training models with inventory information

	All Drugs	HL	HH	LL	LH
LR	0.30 ± 0.03	0.28 ± 0.04	0.33 ± 0.20	0.66 ± 0.21	0.87 ± 0.19
LR_inv	0.29 ± 0.03	0.28 ± 0.04	0.31 ± 0.21	0.65 ± 0.20	0.85 ± 0.19
LR_4	0.30 ± 0.04	0.28 ± 0.04	0.36 ± 0.22	0.65 ± 0.22	0.88 ± 0.22
LR_4_inv	0.30 ± 0.03	0.28 ± 0.04	0.36 ± 0.21	0.64 ± 0.22	0.86 ± 0.20
LR_ATC	0.29 ± 0.03	0.28 ± 0.04	0.35 ± 0.21	0.68 ± 0.22	0.84 ± 0.20
LR_ATC_inv	0.29 ± 0.03	0.28 ± 0.03	0.37 ± 0.20	0.69 ± 0.18	0.91 ± 0.21
LR_DTW	0.29 ± 0.03	0.28 ± 0.04	0.34 ± 0.19	0.67 ± 0.21	0.81 ± 0.16
LR_DTW_inv	0.29 ± 0.04	0.28 ± 0.03	0.31 ± 0.18	0.68 ± 0.22	0.83 ± 0.18
RF	0.29 ± 0.04	0.28 ± 0.05	0.29 ± 0.18	0.72 ± 0.26	0.80 ± 0.16
RF_inv	0.34 ± 0.04	0.29 ± 0.05	0.30 ± 0.17	0.73 ± 0.26	0.91 ± 0.14
RF_4	0.32 ± 0.05	0.30 ± 0.06	0.33 ± 0.22	0.75 ± 0.26	0.83 ± 0.17
RF_4_inv	0.35 ± 0.05	0.31 ± 0.05	0.32 ± 0.21	0.74 ± 0.27	0.82 ± 0.15
RF_ATC	0.34 ± 0.06	0.31 ± 0.07	0.31 ± 0.19	1.21 ± 0.31	3.54 ± 0.27
RF_ATC_inv	0.35 ± 0.07	0.32 ± 0.07	0.33 ± 0.21	1.19 ± 0.28	3.51 ± 0.29
SVR	0.32 ± 0.05	0.30 ± 0.06	0.36 ± 0.19	2.25 ± 1.27	2.15 ± 0.41
SVR_inv	0.41 ± 0.06	0.35 ± 0.05	0.36 ± 0.19	2.21 ± 1.32	2.23 ± 0.38
SVR_4	0.31 ± 0.05	0.30 ± 0.06	0.41 ± 0.23	0.70 ± 0.24	1.17 ± 0.23
SVR_4_inv	0.30 ± 0.05	0.28 ± 0.05	0.29 ± 0.21	0.92 ± 0.25	1.15 ± 0.19
SVR_ATC	0.31 ± 0.04	0.29 ± 0.05	0.29 ± 0.40	1.23 ± 0.42	1.15 ± 0.20
SVR_ATC_inv	0.32 ± 0.05	0.29 ± 0.04	0.28 ± 0.38	0.98 ± 0.42	1.17 ± 0.21
FC	0.32 ± 0.03	0.29 ± 0.04	0.35 ± 0.18	4.96 ± 1.27	4.37 ± 0.53
FC_inv	0.32 ± 0.03	0.29 ± 0.03	0.30 ± 0.17	2.48 ± 1.32	0.84 ± 0.51
FC_4	0.33 ± 0.03	0.30 ± 0.05	0.58 ± 0.30	1.41 ± 0.54	0.87 ± 0.18
FC_4_inv	0.33 ± 0.04	0.30 ± 0.04	0.32 ± 0.31	1.59 ± 0.49	0.91 ± 0.19
FC_ATC	0.32 ± 0.05	0.30 ± 0.06	0.31 ± 0.23	2.66 ± 0.45	0.92 ± 0.16
FC_ATC_inv	0.33 ± 0.05	0.29 ± 0.06	0.30 ± 0.21	2.44 ± 0.45	0.90 ± 0.17
RNN	0.25 ± 0.04	0.26 ± 0.05	0.27 ± 0.17	0.87 ± 0.27	1.28 ± 0.21
RNN_inv	0.25 ± 0.05	0.24 ± 0.06	0.30 ± 0.18	0.84 ± 0.28	1.05 ± 0.21
RNN_4	0.25 ± 0.05	0.24 ± 0.07	0.28 ± 0.19	0.56 ± 0.12	0.65 ± 0.14
RNN_4_inv	0.24 ± 0.05	0.23 ± 0.06	0.27 ± 0.17	0.51 ± 0.19	0.62 ± 0.15
RNN_ATC	0.25 ± 0.05	0.24 ± 0.08	0.26 ± 0.17	0.78 ± 0.31	0.71 ± 0.20
RNN_ATC_inv	0.24 ± 0.05	0.23 ± 0.06	0.25 ± 0.17	0.72 ± 0.25	1.18 ± 0.23
RNN_DTW	0.24 ± 0.05	0.23 ± 0.06	0.27 ± 0.17	0.78 ± 0.20	0.90 ± 0.19
RNN_DTW_inv	0.24 ± 0.04	0.23 ± 0.05	0.27 ± 0.16	0.71 ± 0.26	1.32 ± 0.22

Table 10: Forecasting bias of cross-drug training models with inventory information

	All Drugs	HL	HH	LL	LH
LR	-0.04 ± 0.02	-0.02 ± 0.03	-0.04 ± 0.11	-0.02 ± 0.03	-0.06 ± 0.04
LR_inv	-0.04 ± 0.02	-0.01 ± 0.02	-0.03 ± 0.10	-0.11 ± 0.03	-0.32 ± 0.02
LR_4	-0.03 ± 0.02	-0.02 ± 0.03	-0.02 ± 0.28	-0.16 ± 0.11	-0.56 ± 0.16
LR_4_inv	-0.03 ± 0.02	-0.01 ± 0.02	-0.03 ± 0.25	-0.09 ± 0.13	-0.43 ± 0.16
LR_ATC	-0.02 ± 0.02	-0.01 ± 0.02	-0.04 ± 0.04	-0.08 ± 0.11	-0.17 ± 0.10
LR_ATC_inv	-0.02 ± 0.03	-0.01 ± 0.02	-0.02 ± 0.05	-0.07 ± 0.10	-0.29 ± 0.10
LR_DTW	-0.01 ± 0.02	-0.01 ± 0.02	-0.05 ± 0.03	-0.07 ± 0.05	-0.28 ± 0.13
LR_DTW_inv	-0.01 ± 0.02	0.01 ± 0.02	-0.04 ± 0.03	0.12 ± 0.04	-0.31 ± 0.10
RF	0.01 ± 0.04	0.01 ± 0.05	-0.06 ± 0.05	0.15 ± 0.12	-0.14 ± 0.09
RF_inv	0.04 ± 0.05	0.02 ± 0.04	-0.03 ± 0.06	0.13 ± 0.10	-0.16 ± 0.11
RF_4	0.03 ± 0.06	0.01 ± 0.09	-0.02 ± 0.05	0.12 ± 0.22	-0.07 ± 0.07
RF_4_inv	0.04 ± 0.06	0.01 ± 0.07	-0.02 ± 0.04	0.11 ± 0.18	-0.06 ± 0.06
RF_ATC	0.03 ± 0.07	0.01 ± 0.08	-0.05 ± 0.04	0.13 ± 0.19	1.33 ± 0.08
RF_ATC_inv	0.04 ± 0.07	0.02 ± 0.08	-0.02 ± 0.03	1.21 ± 0.17	0.92 ± 0.07
SVR	0.03 ± 0.06	0.02 ± 0.07	-0.10 ± 0.08	2.02 ± 1.25	1.49 ± 0.50
SVR_inv	0.04 ± 0.07	0.02 ± 0.06	0.06 ± 0.07	0.98 ± 1.21	1.21 ± 0.45
SVR_4	0.02 ± 0.06	0.01 ± 0.07	0.10 ± 0.07	1.01 ± 0.17	-0.13 ± 0.23
SVR_4_inv	0.02 ± 0.06	0.02 ± 0.06	0.03 ± 0.06	0.07 ± 0.15	-0.11 ± 0.21
SVR_ATC	0.03 ± 0.04	0.02 ± 0.05	0.05 ± 0.08	0.08 ± 0.35	2.09 ± 0.16
SVR_ATC_inv	0.01 ± 0.04	0.01 ± 0.04	0.04 ± 0.07	1.23 ± 0.32	2.04 ± 0.14
FC	0.02 ± 0.02	0.01 ± 0.03	-0.04 ± 0.05	4.89 ± 1.16	3.94 ± 0.52
FC_inv	0.02 ± 0.02	0.01 ± 0.02	-0.04 ± 0.04	1.43 ± 1.23	0.32 ± 0.47
FC_4	0.00 ± 0.04	0.02 ± 0.04	-0.26 ± 0.32	0.57 ± 0.70	-0.10 ± 0.32
FC_4_inv	0.01 ± 0.03	0.01 ± 0.04	-0.13 ± 0.31	0.28 ± 0.72	0.42 ± 0.28
FC_ATC	0.02 ± 0.05	0.02 ± 0.06	-0.08 ± 0.07	1.08 ± 0.37	-0.31 ± 0.20
FC_ATC_inv	0.03 ± 0.05	0.02 ± 0.05	-0.07 ± 0.06	0.87 ± 0.39	-0.22 ± 0.21
RNN	0.01 ± 0.04	0.01 ± 0.05	-0.05 ± 0.03	0.17 ± 0.23	0.50 ± 0.12
RNN_inv	0.03 ± 0.04	0.03 ± 0.05	-0.02 ± 0.03	0.19 ± 0.40	0.40 ± 0.17
RNN_4	0.00 ± 0.06	0.00 ± 0.07	-0.04 ± 0.04	0.13 ± 0.14	0.01 ± 0.08
RNN_4_inv	0.02 ± 0.04	0.02 ± 0.05	-0.04 ± 0.03	0.12 ± 0.11	-0.02 ± 0.11
RNN_ATC	0.02 ± 0.05	0.02 ± 0.08	-0.02 ± 0.04	0.22 ± 0.03	0.46 ± 0.03
RNN_ATC_inv	0.01 ± 0.04	0.01 ± 0.05	-0.02 ± 0.03	0.13 ± 0.18	0.45 ± 0.19
RNN_DTW	0.02 ± 0.04	0.03 ± 0.06	-0.06 ± 0.04	0.14 ± 0.03	0.38 ± 0.03
RNN_DTW_inv	0.01 ± 0.04	0.01 ± 0.05	-0.06 ± 0.04	0.11 ± 0.16	0.06 ± 0.07

.2 Performance of Cross-drug Forecasting Models on the Second Dataset

Table 11: Forecasting performance of cross-drug training models with grouping schemes

		Best Baseline	LR	LR_4	LR_ATC	LR_DTW	RNN	RNN_4	RNN_ATC	RNN_DTW
NMSE	All Drugs	4.29	3.98	4.02	4.24	4.23	2.69	2.34	2.43	2.18
	HL	1.34	1.23	1.25	1.32	1.32	0.81	0.73	0.75	0.69
	HH	4.05	4.03	3.96	4.11	3.97	3.58	2.17	2.65	2.06
	LL	0.98	0.96	0.96	0.99	0.98	1.39	0.34	1.42	0.31
	LH	5.12	4.40	4.37	4.34	5.47	4.82	3.66	14.35	3.67
NMAE	All Drugs	0.41	0.40	0.40	0.41	0.41	0.30	0.25	0.30	0.23
	HL	0.38	0.37	0.37	0.38	0.38	0.25	0.23	0.26	0.22
	HH	0.54	0.54	0.54	0.54	0.52	0.53	0.34	0.45	0.32
	LL	0.53	0.53	0.52	0.53	0.53	0.64	0.25	0.64	0.23
	LH	0.59	0.59	0.56	0.54	0.55	0.64	0.37	0.71	0.37
bias	All Drugs	-0.11	-0.07	-0.06	-0.12	-0.12	-0.01	0.00	-0.01	0.00
	HL	-0.10	-0.07	-0.05	-0.11	-0.12	0.00	0.00	0.00	0.00
	HH	-0.16	-0.07	-0.09	-0.17	-0.15	-0.06	-0.02	-0.04	-0.02
	LL	-0.15	-0.04	-0.09	-0.15	-0.16	0.03	-0.01	0.06	-0.01
	LH	-0.13	-0.06	-0.11	-0.15	-0.13	-0.06	-0.05	0.05	-0.04

Table 12: Forecasting performance of RNN models with and without inventory information

		Best Baseline	RNN	RNN_inv	RNN_4	RNN_4_inv	RNN_ATC	RNN_ATC_inv	RNN_DTW	RNN_DTW_inv
NMSE	All Drugs	4.29	2.69	2.14	2.34	1.69	2.43	2.45	2.18	1.39
	HL	1.34	0.81	0.65	0.73	0.51	0.75	0.74	0.69	0.61
	HH	4.05	3.58	2.54	2.17	2.04	2.65	2.87	2.06	1.64
	LL	0.98	1.39	1.35	0.34	0.65	1.42	8.34	0.31	0.29
	LH	5.12	4.82	4.37	3.66	3.87	14.35	5.46	3.67	3.97
NMAE	All Drugs	0.41	0.30	0.26	0.25	0.24	0.30	0.30	0.23	0.22
	HL	0.38	0.25	0.22	0.23	0.22	0.26	0.22	0.22	0.21
	HH	0.54	0.53	0.44	0.34	0.34	0.45	0.35	0.32	0.29
	LL	0.53	0.64	0.62	0.25	0.45	0.64	0.61	0.23	0.22
	LH	0.59	0.64	0.61	0.37	0.44	0.71	0.57	0.37	0.38
bias	All Drugs	-0.11	-0.01	-0.02	0.00	0.00	-0.01	-0.01	0.00	-0.03
	HL	-0.10	0.00	-0.01	0.00	0.00	0.00	-0.01	0.00	-0.03
	HH	-0.16	-0.06	-0.04	-0.02	-0.02	-0.04	-0.04	-0.02	-0.05
	LL	-0.15	0.03	0.05	-0.01	-0.02	0.06	0.14	-0.01	0.00
	LH	-0.13	-0.06	-0.06	-0.05	-0.05	0.05	0.00	-0.04	-0.04

Table 13: Improvement of cross-drug training models with grouping based on volume and variance

		Best Baseline	LR	LR_4	RNN	RNN_4
NMSE	All Drugs	4.29	7.23%	6.29%	37.30%	45.45%
	HL	1.34	8.21%	6.72%	39.55%	45.52%
	HH	4.05	0.49%	2.22%	11.60%	46.42%
	LL	0.98	2.04%	2.04%	--	65.31%
	LH	5.12	14.06%	14.65%	--	28.52%
NMAE	All Drugs	0.41	2.44%	2.44%	26.83%	39.02%
	HL	0.38	2.63%	2.63%	34.21%	39.47%
	HH	0.54	0.00%	0.00%	1.85%	37.04%
	LL	0.53	0.00%	1.89%	--	52.83%
	LH	0.59	0.00%	5.08%	--	37.29%

Table 14: Improvement of cross-drug training models with grouping based on ATC code

		Best Baseline	LR	LR_ATC	RNN	RNN_ATC
NMSE	All Drugs	4.29	7.23%	1.17%	37.30%	43.36%
	A	4.44	3.15%	0.45%	65.77%	81.53%
	B	6.12	3.76%	1.33%	41.67%	72.71%
	C	1.88	9.57%	2.66%	36.17%	43.62%
	J	5.73	--	--	--	8.81%
	M	1.60	0.00%	--	22.50%	42.50%
	N	4.43	--	0.45%	--	8.80%
NMAE	All Drugs	0.41	2.44%	0.00%	26.83%	26.83%
	A	0.47	2.13%	2.13%	17.02%	29.79%
	B	0.65	6.15%	3.08%	21.54%	44.62%
	C	0.37	5.41%	0.00%	32.43%	37.84%
	J	0.51	--	--	--	17.65%
	M	0.43	0.00%	--	6.98%	16.28%
	N	0.51	--	3.92%	--	9.80%

.3 Questionnaire of Pharma Forecasting Practices

1. Your organization is

- □ Retailer/Pharmacy
- □ Pharmaceutical distributor
- □ Pharmaceutical manufacturer
- □ Other, please specify: _____

2. Which group has primary responsibility for the demand forecasting process in your [organization/division]?

- □ Manufacturing
- □ Supply Chain
- □ Marketing or Sales
- □ Finance
- □ Multiple groups have responsibility for their own forecasting process
- □ A functionally independent forecasting group
- □ Other, please specify: _____

3. Which of the following best describes demand forecasts in your [organization/division]?

- □ Made by a statistical software package without involving human judgement.
- □ Based on a statistical forecast, but adjusted by human judgement.
- □ Based entirely on human judgment.
- □ Other, please specify: _____

4. What statistical methods/models do you (or your software) use primarily for preparing forecast?

- □ Moving average

□ Exponential smoothing

□ Regression models

□ Machine smoothing

□ I don't know

□ Other, please specify: _____

5. Does your statistical forecast model (software) incorporate the following information into your demand forecasts?

□ Historical sales

□ Seasonality

□ I don't know what goes into our statistical forecasts

□ Other, please specify: _____

6. a) If human judgement is involved in your demand forecast, how frequent is that?

□ Rarely (< 10% of the time)

□ Sometimes ($10 - 30$% of the time)

□ Very often ($30 - 60$% of the time)

□ Extremely often (> 60% of the time)

b) If human judgement is incorporated into demand forecast, to what extent does human judgement incorporate the following information into your demand forecasts?

	Not at all	Some	A great deal
Planned promotions	□	□	□
New product launches	□	□	□
Other, please specify: _____	□	□	□

7. How often are the demand forecasts generated?

□ Daily

- ☐ Weekly
- ☐ Monthly
- ☐ Quarterly
- ☐ Yearly
- ☐ Other, please specify: _____

8. How much historical data do you use for generating demand forecast?

- ☐ Last 3 months
- ☐ Last 6 months
- ☐ Last year
- ☐ Last 2 years
- ☐ More than last two years
- ☐ Other, please specify: _____

9. How far ahead do you generate your demand forecast?

- ☐ One week
- ☐ Two weeks
- ☐ One month
- ☐ Two months
- ☐ Three months
- ☐ Other, please specify: _____

10. Does your [organization/division] use the demand forecast in any of the following decisions?

	Yes	No	Don't know
Inventory orders	☐	☐	☐
Capacity plans	☐	☐	☐
Cash flow plans	☐	☐	☐
Workforce plans	☐	☐	☐
Production plans	☐	☐	☐
Sales target	☐	☐	☐
Demand/promotion plan	☐	☐	☐
Other, please specify: _____	☐	☐	☐

11. Demand forecasts can be made directly for a hierarchical level, or they can be made indirectly by summing up lower-levels or breaking up a higher level. At which of the following levels does your [organization/division] make demand forecasts directly?

	Yes	No	Don't know
SKU level	☐	☐	☐
Customer level	☐	☐	☐
Store level	☐	☐	☐
Distribution center level	☐	☐	☐
Regional level	☐	☐	☐
National level	☐	☐	☐
Product family level	☐	☐	☐
Other, please specify: _____	☐	☐	☐

12. More advanced models (e.g., machine learning) have been complained about their interpretability (as compared to say, linear regression). While there has been much progress in the interpretability of these models, does the following sentence reflect your [organization/division]'s attitude toward these models? "As long as the more advanced models demonstrate significant accuracy improvement (e.g., $> 10\%$), we are open to using them."

☐ Yes

□ No

□ Other, please specify: _____