Short paper

Accurate Demand Forecasting for Retails with Deep Neural Networks

Shanhe Liao
Tongji University, Shanghai, China
shhliao@tongji.edu.cn

Xianghong Zhou
Tongji University, Shanghai, China
xhz7@tongji.edu.cn

Weixiong Rao
Tongji University, Shanghai, China
wxrao@tongji.edu.cn

ABSTRACT

Demand forecasting, which aims to predict future product sales, is one of the most important tasks in retail markets. With the help of time series prediction models, the literature works either perform the prediction of each individual product item separately, or adopt a multivariate time series forecasting approach. However, none of them leveraged the structural information of product items, such as product brands and multi-level categories. Moreover, various product items have significantly different temporal characteristics, such as periodicity. In this short paper, we propose a deep learning-based prediction model to find inherent inter-dependencies and temporal characteristics among product items for more accurate prediction. Evaluation on two real-world datasets validates that our model can achieve much higher accuracy compared with state-of-the-art methods.

1 INTRODUCTION

Demand forecasting, which aims to predict future product sales, is one of the most important tasks in retail markets. The accurate prediction is important to avoid either insufficient or excess inventory in product warehouses for a typical retail market which typically sells at least thousands of product items.

Traditional works adopt either univariate time series models or multivariate time series model. The univariate time series models, such as the autoregressive integrated moving average (ARIMA) [1], autoregression (AR) [1], moving average (MA) [1] and autoregressive moving average (ARMA) [1], treat different product items separately. ARIMA is rather time consuming especially when there are thousands of products or more. In addition, ARIMA assumes that the current value of time series is a linear combination of historical observations of itself and a random noise. It is hard for ARIMA to capture non-linear relationships and inter-dependencies of different product items. Some machine learning models can also be applied to demand forecasting problems, such as linear regression [11] and linear support vector regression (SVR) [2]. Nonetheless, these machine learning models suffer from the similar weaknesses as ARIMA [15].

Multivariate time series models instead take into account the inter-dependencies among product items. For example, as an extension of ARIMA, vector autoregression (VAR) [9] can handle multivariate time series. However, the model capacity of VAR grows linearly over temporal window size and quadratically over the number of variables, making it hard to model thousands of product items with a long history. More recently, deep learning models have demonstrated outstanding performance in time series forecasting problems. There are basic recurrent neural network (RNN) [5] and its variants including long short-term memory (LSTM) network [10] and gated recurrent unit (GRU) [4].

On this basis, a recent work LSTNet [7] combines convolutional neural network (CNN) [8] and GRU to perform multivariate time series forecasting. The LSTNet model uses a special recurrent-skip component to capture very long-term periodic patterns. However, it assumes that all variables in the multivariate time series have same periodicity, which is invalid for most real datasets.

Other than the aforementioned weaknesses, the existing prediction methods ignore that product items have inherent structural information, e.g., the relations between product items and brands, and the relations among various product items (which may share the same multi-level categories). Our work is motivated by a clustering algorithm [3] to segment product items with help of a so-called *product tree*. This tree structure takes product categories as internal nodes and product items as leaf nodes. Beyond that, we extend the product tree by incorporating product brands and then construct a *product graph* structure. This structure explicitly represents the structural information of product items. Figure 1 illustrates an example of the graph structure of four product items. We can easily find that the brand *Master Kong* has three products, which belong to two different subcategories. Consider that a customer prefers the brand *Master Kong* and recently bought a product item *Master Kong Jasmine Tea*. It is reasonable to infer that he will try another product item *Master Kong Black Tea*, especially when *Master Kong Black Tea* involves a sale promotion campaign.

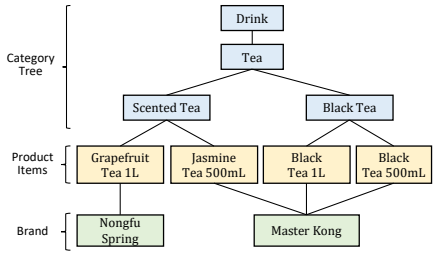

Figure 1: An example of Product Graph Structure involving Categories, Items and Brands.

Without the product graph structure as prior, previous methods either treat all product items equally or have to implicitly infer the inherent relationship but at the cost of accuracy loss. To overcome the issues, with help of the graph structure, we propose a new deep learning model to precisely predict product demands in a multivariate time series forecasting manner, called Structural Temporal Attention Network (STANet). This network incorporates both the product graph structure (see Figure 1) and temporal characteristics of product items. In particular, we note that the inter-dependencies of products and temporal dependencies (e.g., temporal periodicity) may change over time. Thus, we leverage attention mechanism [12] to deal with these variations. In this way, STANet assigns various weights with respect to different inputs involving the variations. Based on graph attention network (GAT) [13], GRU, and a special temporal attention, STANet

performs better than existing methods. As a summary, we make the following contributions.

- We give adequate analysis on two real datasets to validate the motivation of our model, including the product structural inter-dependencies and temporal dependencies.
- The proposed STANet leverages GAT to capture the product structural inter-dependencies and GRU to capture temporal patterns. Moreover, a temporal attention mechanism is adopted on the hidden states of GRU to deal with diverse temporal characteristics of product items. Thus, the two attention mechanisms, i.e., GAT and temporal attention, work together to comfortably learn the product structural inter-dependencies and temporal dependencies.
- Evaluations on two real-world sales datasets show that STANet achieves the best results compared with several state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 gives the problem formulation, and Section3 describes the proposed approach STANet. Then, Section 4 reports evaluation results on two real-world datasets. Finally, Section 5 concludes the paper.

2 PROBLEM FORMULATION

We consider a data set of transaction records in a market. Each transaction record contains 3 fields: the transaction timestamp, item ID, and amount of sold items. Moreover, via the item ID, we can find a list of product categories (in our dataset, each product item is with a list of 4-level categories) and an associated product brand. In this way, we augment each transaction record by totally 8 (=3+4+1) fields. Given a certain time horizon (e.g., one day or one week), we pre-process the transaction records into a multivariate time series of the volumes of sold product items. In addition, for a certain category (or brand), we sum the volumes of all product items belonging to the category (or brand). In this way, we have the multivariate time series of the volumes of product items, categories, and brands. Meanwhile, the product graph structure is stored in an adjacency matrix, where an element 1 indicates an edge between two nodes (such as a product item and its brand), otherwise 0.

Formally, we denote the number of product items by N_p and the total number of items, brands and categories as N. Given the augmented multivariate time series $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$, $\boldsymbol{x}_t \in \mathbb{R}^{N\times 1}$, $t = 1, 2, \ldots, T$ and adjacency matrix $M \in \mathbb{R}^{N\times N}$, we aim to predict future product sale volume $\boldsymbol{x}_{T+h}$ where h is the desirable horizon ahead of the current time stamp. More specifically, to train a model using historical data, we use a time window of size τ to split existing data into fixed length inputs, where each input is expressed as $\{\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t+\tau-1}\}$ and $\boldsymbol{x}_{t+\tau-1+h}$ is the label. The adjacency matrix M is fixed. In this way, the demand forecasting problem is equivalent to learning a function $f_M : \mathbb{R}^{N\times\tau} \rightarrow \mathbb{R}^{N\times 1}$. On the testing stage, we only need to calculate evaluation metric for the N_p real product items.

3 FRAMEWORK

In this section, we present the detail of the proposed model STANet. Figure 2 gives the framework of STANet.

3.1 Graph Attention Component

For multivariate time series forecasting, one of the most crucial tasks is to capture the inter-dependencies between different variables. What's more, as shown in Section 4.1, the inter-dependencies may change over time. To explore inter-dependencies

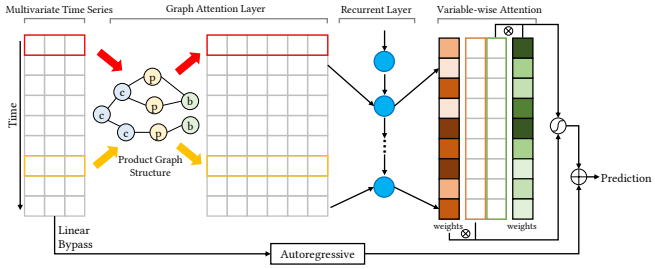

Figure 2: The framework of STANet.

from product graph structure, we need graph neural network. Furthermore, to handle dynamic inter-dependencies, we choose attention mechanism because it can assign various weights with respect to different inputs. As a result, the first layer of STANet is a graph attention layer [13]. Given the input time series $X \in \mathbb{R}^{N\times\tau}$ and adjacency matrix $M \in \mathbb{R}^{N\times N}$, we use a multi-head graph attention layer to process X time step by time step. Formally, this operation at time step t is given by

$$\boldsymbol{h}_t^i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k W^k x_t^j\right), \tag{1}$$

where x_t^j is the sale of variable (product, or brand, or category) j at time step t, W^k is a linear transformation to obtain sufficient expressive power, $\mathcal{N}_i$ refers to all the adjacent nodes of variable i given by M, K is the number of multi-head attention and σ is an activation function. α_{ij}^k is the coefficient of attention mechanism computed by

$$\alpha_{ij}^k = \frac{exp\left(LeakyReLU\left(f_a\left(W^k x_t^i, W^k x_t^j\right)\right)\right)}{\sum_{\ell\in\mathcal{N}_i} exp\left(LeakyReLU\left(f_a\left(W^k x_t^i, W^k x_t^\ell\right)\right)\right)}, \tag{2}$$

where f_a is a scoring function to evaluate relevance, and in our model it is a single-layer feedforward neural network.

Suppose $W^k \in \mathbb{R}^{F\times 1}$, with aforementioned X and M, the output of the graph attention component is $X_G \in \mathbb{R}^{FN\times\tau}$.

3.2 Recurrent Component

When the variable-to-variable relationships have been processed, X_G is fed into the recurrent component to capture temporal patterns. Here we use gated recurrent unit (GRU) [4] as the recurrent layer. Compared with vanilla recurrent neural networks (RNN) [5], GRU is more capable to capture long-term patterns. Suppose the hidden size of GRU is d_r, then the output of recurrent component is $X_R \in \mathbb{R}^{d_r\times\tau}$.

3.3 Variable-Wise Temporal Attention

After the graph attention component and recurrent component, the model has successfully captured inter-dependencies and basic temporal patterns. Nonetheless, temporal patterns could also be dynamic. Therefore, as a commonly used technique in RNN, temporal attention can be added to the model as

$$\boldsymbol{\alpha}_{t+\tau-1} = f_a\left(H_{t+\tau-1}, \boldsymbol{h}_{t+\tau-1}\right), \tag{3}$$

where $\boldsymbol{\alpha}_{t+\tau-1} \in \mathbb{R}^{\tau\times 1}$, f_a is a scoring function and $\boldsymbol{h}_{t+\tau-1}$ is the last hidden state of RNN, $H_{t+\tau-1} = [\boldsymbol{h}_t, \ldots, \boldsymbol{h}_{t+\tau-1}]$ is a matrix stacking the hidden states of RNN.

However, we will show in Section 4.1 that various products may have rather different temporal characteristics such as periodicity. Instead of using the same attention mechanism for all product items, we propose a variable-wise temporal attention

Table 1: Datasets statistics, where T is length of time step, P is the time interval, N_p, N_b, N_c are numbers of products, brands, categories respectively, $N = N_p + N_b + N_c$, and sparsity means proportion of zero value in the data.

Datasets	T	P	N_p	N_b	N_c	N	Sparsity
Dataset-1	572	1 day	1878	433	612	2923	53%
Dataset-2	833	1 day	1925	289	771	2985	39%

mechanism to compute the attention for each variable independently as

$$\boldsymbol{\alpha}^i_{t+\tau-1} = f_a\left(H^i_{t+\tau-1}, \boldsymbol{h}^i_{t+\tau-1}\right). \tag{4}$$

Equation (4) is similar to Equation (3), except a superscript $i = 1, 2, \ldots, d_r$, indicating that the attention mechanism is calculated for a particular GRU hidden variable. In this way, our model could deal with different temporal characteristics such as periodicity for different variables. With the coefficients $\boldsymbol{\alpha}^i_{t+\tau-1}$, the weighted context vector of i^{th} hidden variable is calculated as

$$\boldsymbol{c}^i_{t+\tau-1} = H^i_{t+\tau-1}\boldsymbol{\alpha}^i_{t+\tau-1}, \tag{5}$$

where $H^i_{t+\tau-1} \in \mathbb{R}^{1\times\tau}$ and $\boldsymbol{\alpha}^i_{t+\tau-1} \in \mathbb{R}^{\tau\times 1}$. Let $\boldsymbol{c}_{t+\tau-1}$ be context vector of all hidden variables, then we can calculate the final output for horizon h as

$$\boldsymbol{y}_{t+\tau-1+h} = W\left[\boldsymbol{c}_{t+\tau-1}; \boldsymbol{h}_{t+\tau-1}\right] + b, \tag{6}$$

here $W \in \mathbb{R}^{N\times 2d_r}$ and $b \in \mathbb{R}^{1\times 1}$ are parameters of a fully connected layer.

3.4 Autoregressive Component

Similar to LSTNet [7], we add an autoregressive component to capture the local trend of product demands. This component is a linear bypass that predicts future demands directly from the input data to address the scale problem. This linear bypass will fit all products' historical data with a single linear layer.

The final prediction of STANet is then obtained by integrating the outputs of the neural network part and the autoregressive component using an automatically learned weight.

4 EXPERIMENTS AND EVALUATIONS

We first analyze two used real-world datasets to motivate STANet and then compare STANet against 6 counterparts.

4.1 Datasets and Analysis

We use two real-world datasets collected from two medium size stores of a chain retail in Shandong Province, China. Table 1 summarizes the statistics. As shown in this table, the sparsity of dataset-1 is greater than 50%, which makes the forecasting task rather challenging. Both datasets are split into training set (70%), validation set (15%) and testing set (15%) in chronological order. To explore the inter-dependencies and temporal characteristics of datasets, we give following analysis.

We consider two variables have inter-dependencies if the history of one variable can help forecasting another. Assuming two univariate time series $\boldsymbol{x} = x_1, x_2, \ldots, x_T$, $\boldsymbol{y} = y_1, y_2, \ldots, y_T$ and a specific time lag m, we model $\boldsymbol{y}$ as a regression of itself and $\boldsymbol{x}$:

$$y_t = a_0 + a_1 y_{t-1} + \ldots + a_m y_{t-m} + b_1 x_{t-1} + \ldots + b_m x_{t-m}. \tag{7}$$

If $\boldsymbol{y}$ gets the best fitting when all $b_i = 0$ for $i = 1, \ldots, m$, we believe that $\boldsymbol{x}$ cannot help forecasting $\boldsymbol{y}$. To test whether $\boldsymbol{x}$ can help forecast $\boldsymbol{y}$, we use the Granger causality test [6], and for each lag m the result maybe differ. We use

$$GR_{\boldsymbol{x},\boldsymbol{y}} = \frac{number\ of\ m\ where\ \boldsymbol{x}\ helps\ forecast\ \boldsymbol{y}}{total\ number\ of\ m} \tag{8}$$

to represent importance of $\boldsymbol{x}$ to $\boldsymbol{y}$, $GR_{\boldsymbol{x},\boldsymbol{y}} \in [0, 1]$. We select one category and two concrete products to verify the inter-dependencies in Figure 3.

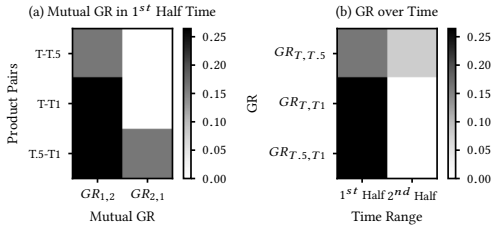

Figure 3: GR among three variables, where $T.5$ is the product item "Black Tea 500mL", $T1$ is the item "Black Tea 1L" and T is the category "black tea". Time range is split into two parts. (a) Mutual GR of three pairs. (b) Change of dependencies over time.

From Figure 3(a), we can find that $GR_{T,T.5}$ and $GR_{T,T1}$ are quite darker while $GR_{T.5,T}$ and $GR_{T1,T}$ are almost white. This figure means that the historical data of the category "black tea" helps forecasting the demand of its children and instead the children's history data is trivial to the forecast of their parent category. $GR_{T.5,T1}$ and $GR_{T1,T.5}$ involve the different degree of darkness. It means that $GR_{T.5,T1}$ can improve the forecast of $GR_{T1,T.5}$ and vice versa. However, the improvement degree is not identical. Figure 3(b) shows that the inter-dependencies is dynamically changing over time.

To verify that various variables have different temporal characteristics, we use Fast Fourier Transform (FFT) to plot the periodogram [14] for the category "black tea" and its two children product items in Figure 4. We can easily find that they have rather different periodicity at the rectangles. Thus, we cannot simply apply the same attention onto all variables.

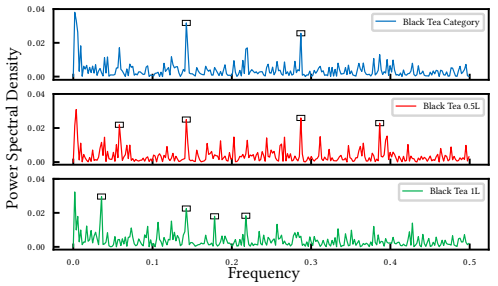

Figure 4: Periodogram of one category and two products. The rectangles highlight points for periodicity candidates.

4.2 Methods and Metric

We compare our model with five other methods, and their variants. For AR, Ridge and LSVR, we train models for each product separately, while neural network models take multivariate data as input directly.

- AR [1]: A classic univariate time series modeling method.
- Ridge [11]: Linear regression with $\ell 2$ regularization.
- LSVR [2]: Linear SVR, another machine leaning method for regression.

Table 2: RSE of six methods and their variants on two datasets: boldface indicates the best result on each dataset.

Methods	Dataset-1	Dataset-2
AR [1]	0.8016	0.5313
Ridge [11]	0.7981	0.5628
LSVR [2]	3.2446	7.8246
GRU [4]	0.8535	0.9067
LSTNet [7]	0.7907	0.7570
LSTNet-IAttn	0.8230	0.5399
STANet	**0.7783**	**0.5233**
STANet-oStructure	0.8907	0.6179
STANet-oCategory	0.8256	0.5569
STANet-oBrand	0.8640	0.5471
STANet-oAR	0.8327	0.8850

- GRU [4]: Recurrent neural network using GRU cell.
- LSTNet [7]: A state-of-the-art model composed of CNN, GRU and highway network.
- LSTNet-IAttn: LSTNet improved by the variable-wise temporal attention.
- STANet-oStructure: STANet without structural information.
- STANet-oCategory: STANet without category information.
- STANet-oBrand: STANet without brand information.
- STANet-oAR: STANet without autoregressive component.

We measure the forecasting performance by root relative squared error (RSE).

$$RSE = \frac{\sqrt{\sum_{i=1}^{N}\sum_{t=t_0}^{t_1}\left(y_{i,t}-\hat{y}_{i,t}\right)^2}}{\sqrt{\sum_{i=1}^{N}\sum_{t=t_0}^{t_1}\left(y_{i,t}-mean(Y)\right)^2}}, \tag{9}$$

where y and $\hat{y}$ are ground truth and predicted value respectively, t_0 and t_1 are start and end time of testing set, and $Y \in \mathbb{R}^{N\times(t_1-t_0)}$ represents the matrix of all labels y in testing set. RSE can be regarded as RMSE divided by standard deviation of testing set, so scale differences between different datasets can be ignored. Lower RSE generally means better forecasting performance.

4.3 Results of Different Methods

Table 2 provides the RSE of aforementioned methods on two datasets for horizon = 1. Our proposed model STANet outperforms others on both datasets. It is because STANet leverages the attention mechanism and inherent product structural information to precisely capture structural and temporal dependencies. LSVR performs worst on both datasets. Vanilla GRU suffers from worse performance than univariate models, because not every product has inter-dependency with each other and simply adding irrelevant data would harm the forecasting task. LSTNet can achieve lower errors than AR and Ridge on dataset-1, but not on dataset-2. It is mainly because the product items in dataset-1 exhibit much stronger structural inter-dependencies than those in dataset-2. For most methods except LSVR and GRU, the RSE on dataset-2 is much lower than that on dataset-1. It is due to the fact that dataset-1 contains much sparser data than dataset-2.

In terms of the ablation experiments of STANet, we can find that the structural information and the AR component play the major contribution in the forecasting task. For example, the result of STANet-oStructure indicates that the removal of structural information could greatly harm the forecasting accuracy. Incorporating the brand and category information will benefit the forecast and their corresponding contribution heavily depends upon the datasets. Both parts work together to the best performance. Also the results of STANet-oAR without the AR component indicate that its RSE is much higher than the original STANet especially on dataset-2. This is because the AR component can more comfortably capture the local trend in dataset-2 with a lower sparsity than the one in dataset-1. Finally, by comparing the results of LSTNet and LSTNet-IAttn, we find that LSTNet-IAttn by incorporating the variable-wise attention mechanism can greatly improve the forecast performance on dataset-2.

5 CONCLUSIONS

In this paper, we propose a novel deep learning-based forecasting model STANet in a multivariate time series manner. The model integrates the four components of GAT, GRU, variable-wise attention mechanism and auto-regression to precisely capture the inherent product structural information and temporal periodicity for more accurate prediction. Our analytic result on two real datasets demonstrates that the two real datasets exhibit strong product structural information and temporal periodicity. The evaluation result on the two datasets validates that STANet outperforms 6 counterparts and 4 variants of STANet. As the future work, we plan to further improve STANet and provide more experimental results on both online and offline transaction data.

ACKNOWLEDGMENTS

This work is partially supported by National Natural Science Foundation of China (Grant No. 61572365, No. 61772371, No. 61972286).

REFERENCES

[1] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
[2] L.-J. Cao and F. E. H. Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks*, 14(6):1506–1518, 2003.
[3] X. Chen, J. Z. Huang, and J. Luo. Purtreeclust: A purchase tree clustering algorithm for large-scale customer transaction data. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 661–672. IEEE, 2016.
[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
[5] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.
[6] M. Gregorova, A. Kalousis, and S. Marchand-Maillet. Learning predictive leading indicators for forecasting time series systems with unknown clusters of forecast tasks. *arXiv preprint arXiv:1710.00569*, 2017.
[7] G. Lai, W.-C. Chang, Y. Yang, and H. Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104. ACM, 2018.
[8] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.
[9] H. Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.
[10] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, page 89. Presses universitaires de Louvain, 2015.
[11] G. A. Seber and A. J. Lee. *Linear regression analysis*, volume 329. John Wiley & Sons, 2012.
[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
[14] M. Vlachos, P. Yu, and V. Castelli. On periodicity detection and structural periodic similarity. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 449–460. SIAM, 2005.
[15] J. Yin, W. Rao, M. Yuan, J. Zeng, K. Zhao, C. Zhang, J. Li, and Q. Zhao. Experimental study of multivariate time series forecasting models. In *ACM CIKM 2019*, pages 2833–2839.