Multiple Stock Time Series Jointly Forecasting with Multi-Task Learning

Tao Ma and Ying Tan
Key Laboratory of Machine Perception (MOE)
Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University
Beijing, 100871, China
Email: {pku_mark, ytan}@pku.edu.cn

***Abstract*—Due to the strong connections among stocks, the information valuable for forecasting is not only included in individual stocks, but also included in the stocks related to them. These inter-correlations can provide invaluable information to be further leveraged to improve the overall forecasting performances. However, most previous works focus on the forecasting task of one single stock, which easily ignore the valuable information in others. Therefore, in this paper, we propose a jointly forecasting approach to process the time series of multiple related stocks simultaneously, using multi-task learning framework. In particular, this framework processes multiple forecasting tasks of different stocks simultaneously by sharing the information extracted based on latent inter-correlations. Meanwhile, each stock has their private encoding networks to keep their own information. Moreover, to dynamically balance private and shared information, we propose an attention based method, called Shared-private Attention, to optimally combine the shared and private information of stocks, which is inspired by the idea of Capital Asset Pricing Model (CAPM). Experimental results on the datasets of both stock and other domains demonstrate the proposed method can outperform other methods in forecasting performance.**

***Index Terms*—stock time series, forecasting, multi-task learning, neural networks**

I. Introduction

Time series forecasting is to build models which can forecast future values based on the past information. This problem widely exists in many real-world scenarios, such as finance, logistic, natural environment, medical analysis, etc. In most cases, the time series we deal with are not univariate but multivariate, so it is also called multivariate time series forecasting. In this paper, we focus on the problem of stock time series forecasting.

Stock time series, are extremely challenging to predict because of their low signal-noise ratio [1] and heavy-tailed distributions [2]. Meanwhile, the predictability of stock market returns still remains open and controversial [3]. To achieve the good forecasting performance, many classic statistical solutions such as MA, AR, VARMA [4], as well as machine learning based methods such as Neural Networks [5] and SVM [6], are proposed to deal with it, yielding encouraging performance. However, most of these methods are focusing on analyzing one single stock. Actually, the information contained in a single stock's time series is often limited. According to the theory of Capital Asset Pricing Model (CAPM) [7], the returns of all individual stocks are affected by the systemic risk, in other words, they are all affected by the macro market. Due to the inter-connections among stocks, a lot of information valuable for forecasting is actually included in the time series of other related ones, not just the individual. When analyzing stocks independently, it is very difficult to capture them all. Thus, it is better to process multiple related stocks at the same time.

To leverage more information from related stocks, a straightforeword solution is Multi-Task Learning (MTL) [8], which is already widely used in text and image applications [9], [10]. MTL jointly learns multiple related tasks and leverages the correlations over tasks to improve the performance. Therefore, it often works better than single-task learning. Some recent works apply MTL to time series forecasting, e.g. the works of [11] and [12]. However, there are some limitations in these approaches: 1) only learn the shared information but ignore the task-private: most of them use a single encoding model to learn the shared latent features of all tasks, which makes it easily ignore the useful task-private information; 2) simply put all latent features together: some other approaches build multiple models to learn both the shared and task-private latent features, but they simply put these features together and feed them to the dense layer, instead of integrating them with more knowledge.

To address the problems of these existing works, in this paper, we propose a multi-series jointly forecasting approach for multiple stocks forecasting, as well as a new attention method to optimally balance the shared and private information. More specifically, in our MTL based method, each task represents the forecasting of a single stock. Only the shared information is not enough, so we build multiple networks to learn both the shared and private latent features from multiple time series of related stocks using MTL. To combine the information with more valuable knowledge, we build an attention model to learn an optimized weighted combination of them inspired by the idea of Capital Asset Pricing Model (CAPM) and Attention [13].

Experimental results on the datasets of both stock and other domains demonstrate the proposed method can outperform the previous works, including classic methods, single-task methods, and other MTL based solutions [14].

Ying Tan is the corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

The contributions of this paper are multifold:

- The proposed multi-series jointly forecasting approach applies multi-task learning framework to time series forecasting for multiple related stocks.
- We propose a attention based method to learn the optimized combination of shared and task-private latent features of stocks, which is inspired by the idea of CAPM.
- We demonstrate in the experiments on real-world stock dataset that the proposed approach outperforms single-task baselines and other MTL based methods, which further improves the forecasting performance.

The remainder of the paper is organized as follows: related works are introduced in Section II. The details of the proposed method are presented in Section III. Experiments on various datasets are demonstrated in Section IV, including the results and analysis. Finally, we conclude in Section V.

II. Related Work

A. Time Series Forecasting

The study of time series forecasting has a long history in the field of economics. Due to its importance to investing, it is still attractive to researchers from many fields, not only economics but also machine learning and data mining.

Many classic linear stochastic models are proposed and widely used, such as AR, ARIMA [15] and GARCH [16]. However, most of these methods pay more attention to the interpretation of the variables than improving the forecasting performance. Especially when dealing with complex time series, they perform poorly. To improve the performance, Gaussian Processes (GP) are used [17], which works better especially when the time series are sampled irregularly [18].

On the basis of these methods yielding to, some works bring in Machine Learning (ML), e.g., the Gaussian Copula Process Volatility model [19], which brings GP and ML together.

With the recent advances of Deep Learning (DL), many promising achievements have been achieved in many applications of machine learning in the past few years [20], such as computer vision [21], natural language processing (NLP) [22], [23] and speech recognition [24]. Recently, many works apply DL to forecasting time series [25]–[27]. However, there are still few works using deep learning for financial forecasting. For some recent examples, [28] applied deep learning to event-driven stock market prediction. [29] used autoencoders with one single layer to compress multivariate financial data. [30] present augmentation of LSTM architecture, which is able to process asynchronous series. [31] proposed autoregressive convolutional neural networks for asynchronous financial time series.

These works have a common limitation: they only focus on the time series of one single stock, or even a univariate time series. Even if they can process multiple time series of multiple stocks, they still don't make good use of the connections among stocks to extract all the information.

B. Deep Multi-task Learning

Multi-task Learning (MTL) is to process multiple related tasks at the same time, leveraging the correlation over tasks to improve the performance. In recent years, it often comes with deep learning, so also called Deep Multi-task Learning (DMTL). Generally, if you find your loss function optimizes multiple targets at the same time, you actually do multi-task learning [32]. It has successfully applied in all applications of machine learning, including natural language process [22] and computer vision [33].

There are some recent works using DMTL to deal with time series forecasting problems. [34] used multi-task Gaussian processes to process physiological time series. [35] proposed a multi-task learning approach to learn the conditional independence structure of stationary time series. [36] used multi-task multi-view learning to predict urban water quality. [11] used recurrent LSTM neural networks and multi-task learning to deal with clinical time series. And [12] applied multi-task representation learning to travel time estimation. Moreover, some methods are proposed to learn the shared representation of all the task-private information, e.g., [37] proposed cross-stitch networks to combine multiple task-private latent features.

There are some limitations in these works. Firstly, most of them ignore the task-private information since they only build a single model to learn the shared information of multiple tasks. Secondly, although some consider the task-private information, they do not make use of them efficiently since they simply put these latent features together and feed them to the forecasting model.

III. Methods

To address the limitations that the previous works focus on a single stock and use the shared information only, we propose a new method based on Deep Multi-Task Learning (DMTL) for financial forecasting. More specifically, to efficiently extract the shared and private information from multiple related stocks, we build multiple networks to learn their latent representations using DMTL. Furthermore, to address the problems of not efficiently combining the shared and private information, we propose an attention method to learn their optimized combination inspired by the idea of CAPM. We will describe the details in the following.

A. Problem Statement

Firstly, we give the formal definition of the time series forecasting problems:

$$\hat{\boldsymbol{y}} = g(\boldsymbol{x}_{t-1}, \ldots, \boldsymbol{x}_{t-N}) \approx \mathrm{E}[\boldsymbol{y}_t | \{\boldsymbol{x}_{t-i},\ i = 1, \ldots, N\}]\,, \quad (1)$$

where $\mathrm{E}(\cdot)$ is the mathematical expectation, $g(\cdot)$ is the approximate function, and N is the length of past sequence.

The time series could be multivariate, that is, $\boldsymbol{x}_t$ represent the values of multiple series at time t:

$$\boldsymbol{x}_t = (x_{1t}, x_{2t}, \ldots, x_{mt})^{\mathrm{T}}\,, \quad (2)$$

where m is the number of series. For example, for a stock, there are multiple price series, such as opening prices, closing

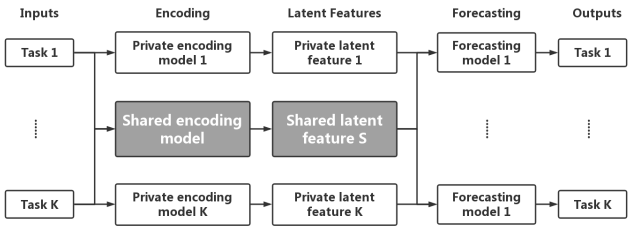

Fig. 1. The architecture of MSJF. It processes the forecasting tasks of K related stocks at the same time. The shared encoding model $\mathrm{enc}_s(\cdot)$ extracts the shared information $\boldsymbol{f}_s$ from all stocks, while each stock has its private encoding model $\mathrm{enc}_k(\cdot)$, extracting their private information $\boldsymbol{f}_k$. Then each of them has private forecasting module $\mathrm{F}_k(\cdot)$, using both the shared and private information to forecast the future values.

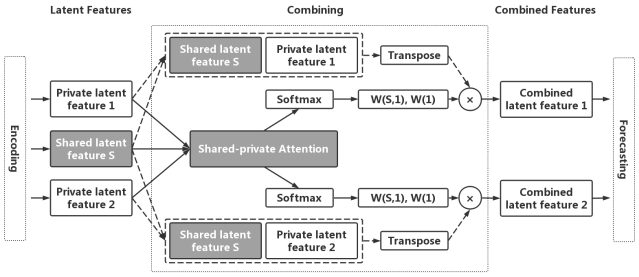

Fig. 2. The architecture of SPA. The encoding models extract the latent features $\boldsymbol{f}_s$ and $\boldsymbol{f}_k$ from the raw time series features. SPA measures the contributions of them to task k and learns their weights, w_{sk} and w_k. Then they are combined with w_{sk} and w_k into the optimized representations $\tilde{\boldsymbol{f}}_k$. The dotted lines represent the duplication of the latent features to be combined by SPA, and the solid lines represent the inputs and outputs of the SPA neural network.

prices and so on. Moreover, $\boldsymbol{y}$ can be $\boldsymbol{x}$ itself, which is called autoregressive forecasting.

Then, for Multi-task Learning, assuming that there are K tasks in total, the problem is defined as:

$$\begin{bmatrix} \hat{\boldsymbol{y}}_1 \\ \vdots \\ \hat{\boldsymbol{y}}_K \end{bmatrix} = g \begin{pmatrix} \{\boldsymbol{x}_{t-i}^1\}_{i=1}^N \\ \vdots \\ \{\boldsymbol{x}_{t-i}^K\}_{i=1}^N \end{pmatrix} \approx \mathrm{E} \left[\begin{array}{c|c} \boldsymbol{y}_t^1 & \{\boldsymbol{x}_{t-i}^1\}_{i=1}^N \\ \vdots & \vdots \\ \boldsymbol{y}_t^K & \{\boldsymbol{x}_{t-i}^K\}_{i=1}^N \end{array}\right], \tag{3}$$

where $\{\boldsymbol{x}_{t-i}^k\}_{i=1}^N = \{\boldsymbol{x}_{t-i}^k \mid i = 1, 2, \ldots, N\}$ is the time series of task k, and N is the length of the time series.

In this paper, we process the forecasting tasks of multiple related stocks, each of which has their own time series for inputs.

B. Multi-series Jointly Forecasting

In order to utilize the connections to extract the valuable information from multiple related stocks and improve the forecasting performance, we propose a jointly forecasting approach based on DMTL to process multiple stocks simultaneously, called Multi-series Jointly Forecasting (MSJF).

According to the theory of CAPM, there are strong connections among stocks. However, these connections are complicated to clearly quantify and describe in the model. If these connections are utilized, the forecasting performance will be further improved. Therefore, we propose MSJF, the framework of which can be found in Figure 1, to leverage the connections among tasks to forecast multiple stocks. Formally, MSJF with K tasks can be defined as:

$$\begin{aligned} \begin{bmatrix} \hat{\boldsymbol{y}}_1 \\ \vdots \\ \hat{\boldsymbol{y}}_K \end{bmatrix} &= \mathrm{MSJF} \begin{pmatrix} \{\boldsymbol{x}_{t-i}^1\}_{i=1}^N \\ \vdots \\ \{\boldsymbol{x}_{t-i}^K\}_{i=1}^N \end{pmatrix}, \\ \hat{\boldsymbol{y}}_k &= \mathrm{F}_k\left(\boldsymbol{f}_s, \boldsymbol{f}_k\right), \ k = 1, 2, \ldots, K, \\ \boldsymbol{f}_s &= \mathrm{enc}_s\left(\|_{k=1}^K \{\boldsymbol{x}_{t-i}^k\}_{i=1}^N\right), \\ \boldsymbol{f}_k &= \mathrm{enc}_k\left(\{\boldsymbol{x}_{t-i}^k\}_{i=1}^N\right), \ k = 1, 2, \ldots, K, \end{aligned} \tag{4}$$

where

- $\hat{\boldsymbol{y}}_k$ is the predicted values.
- $\mathrm{F}_k(\cdot)$ is the forecasting model for stock k, using both the shared and private information.
- $\mathrm{enc}_s(\cdot)$ is the shared encoding model, using the stock correlations to extract the shared information, $\mathrm{enc}_k(\cdot)$ is the private encoding model for stock k, to keep its own information.
- $\|$ is the concatenate operation.
- $\boldsymbol{f}_s$ is the shared information, $\boldsymbol{f}_k$ is the private information of stock k.

Then, MSJF jointly trains all the tasks end-to-end, by following joint loss function:

$$L = \frac{1}{K} \sum_{k=1}^{K} l_k(\boldsymbol{Y}_k, \hat{\boldsymbol{Y}}_k), \tag{5}$$

where L is the joint loss, l_k is the loss function of task k, $\boldsymbol{Y}_k$ is the ground truth of all samples in task k and $\hat{\boldsymbol{Y}}_k$ is the forecasting values of all samples in task k.

C. Shared-private Attention

To combine the shared and task-private latent features with more valuable knowledge, instead of simply putting them together, we propose an attention model to learn the optimized combination of them inspired by the idea of CAPM.

Capital Asset Pricing Model [7]: given an asset (e.g., stock) i, the relationship between its excess earnings and the excess earnings of market can be expressed as:

$$E(r_i) - r_{\mathrm{f}} = \beta_{im} \cdot [E(r_m) - r_{\mathrm{f}}], \tag{6}$$

where

- $E(r_i)$ is the expected return on the capital asset i, $E(r_m)$ is the expected return of the market m.
- r_{f} is the risk-free return, such as interest arising from government bonds.
- β_{im} is the sensitivity of the expected excess return of asset i to the expected excess return of market m.

CAPM suggests that the return of the capital asset can be explained by the return of macro market.

Then, subsequent work [38] shows that there are excess returns in the earnings of the capital asset that exceeds the market benchmark portfolio.

$$R_i - r_{\mathrm{f}} = \beta_{im} \cdot (R_m - r_{\mathrm{f}}) + \alpha_i, \tag{7}$$

where α_i is the excess return of asset i that exceeds the market benchmark portfolio. For stocks, the return of a single stock actually receives varying degrees of influence from the macro market (often called Beta) and its own factors (often called Alpha). And the levels of these influences vary from different stocks. If the levels are expressed by weights, then the return of individual stocks can be described as:

$$R_i - r_{\mathrm{f}} = w_{\mathrm{B}} \cdot R_{\mathrm{Beta}} + w_{\mathrm{A}} \cdot R_{\mathrm{Alpha}} \,. \tag{8}$$

Similarly, in our DMTL model, each task represents a single stock, then it is also influenced by the market (shared information) and its own factors (task-private information), the levels of which can be different and vary from different tasks. So based on this, we aim to combine these information with their levels of influence.

Attention mechanism measures the importance of objects in your vision and learns their importance by weighting them. Therefore, we use an attention model to measure the contributions of the shared information $\boldsymbol{f}_s$ and the task-private information $\boldsymbol{f}_k$ to its own forecasting task k. Then the model combines these information with their weights and obtains the optimized combination, which is called Shared-private Attention (SPA). The relationships between $\boldsymbol{f}_s$ and $\boldsymbol{f}_k$ can be described as follows:

$$\begin{aligned} (w_{sk}, w_k) &= \mathrm{softmax}\left(\mathrm{SPA}\left(\boldsymbol{f}_s \| \boldsymbol{f}_k\right)_s, \mathrm{SPA}\left(\boldsymbol{f}_s \| \boldsymbol{f}_k\right)_k\right) , \\ \tilde{\boldsymbol{f}}_k &= (w_{sk}, w_k) \cdot (\boldsymbol{f}_s, \boldsymbol{f}_k)^{\mathrm{T}} , \ k = 1, 2, \ldots, K \,, \end{aligned} \tag{9}$$

where

- w_{sk} is the weights of shared features $\boldsymbol{f}_s$ for task k.
- w_k is the weights of private features $\boldsymbol{f}_k$ for its own task k.
- $\mathrm{SPA}(\cdot)$ is the attention mechanism computing *attention coefficients*, which is a neural network model.
- $\mathrm{SPA}\left(\boldsymbol{f}_s \| \boldsymbol{f}_k\right)_s$ and $\mathrm{SPA}\left(\boldsymbol{f}_s \| \boldsymbol{f}_k\right)_k$ means the outputs of the attention neural network.
- $\tilde{\boldsymbol{f}}_k$ is the optimized combined latent features.

Finally, MSJF uses the combined latent features $\tilde{\boldsymbol{f}}_k$ to do jointly forecasting, which is called Multi-series Jointly Forecasting with Shared-private Attention (SPA-MSJF).

$$\begin{aligned} \begin{bmatrix} \hat{\boldsymbol{y}}_1 \\ \vdots \\ \hat{\boldsymbol{y}}_K \end{bmatrix} &= \mathrm{MSJF} \begin{pmatrix} \{\boldsymbol{x}_{t-i}^1\}_{i=1}^N \\ \vdots \\ \{\boldsymbol{x}_{t-i}^K\}_{i=1}^N \end{pmatrix} , \\ \hat{\boldsymbol{y}}_k &= \mathrm{F}_k\left(\tilde{\boldsymbol{f}}_k\right), \ k = 1, 2, \ldots, K \,, \\ \boldsymbol{f}_s &= \mathrm{enc}_s\left(\|_{k=1}^K \{\boldsymbol{x}_{t-i}^k\}_{i=1}^N\right) , \\ \boldsymbol{f}_k &= \mathrm{enc}_k\left(\{\boldsymbol{x}_{t-i}^k\}_{i=1}^N\right), \ k = 1, 2, \ldots, K \,, \\ \tilde{\boldsymbol{f}}_k &= (w_{sk}, w_k) \cdot (\boldsymbol{f}_s, \boldsymbol{f}_k)^{\mathrm{T}} , \ k = 1, 2, \ldots, K. \end{aligned} \tag{10}$$

D. Multi-head Shared-private Attention

During the experiments, we found that the learning process of Shared-private Attention is not very stable on the data with strong volatility, for example, financial data. To stabilize its learning process, we apply multi-head attention [39] to our model, which is an ensemble method to combine the outputs of several independent attention models and stabilize the learning process. We call it Multi-head Shared-private Attention (MH-SPA).

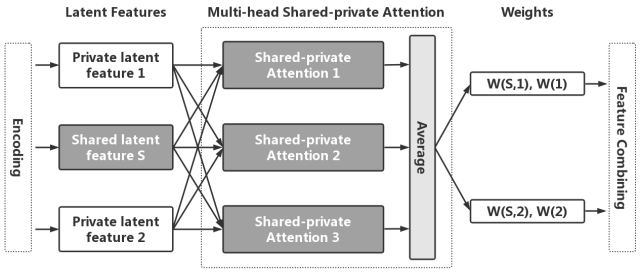

Fig. 3. The architecture of MH-SPA. Several attention models compute the weights with independent parameters in parallel. And their outputs are averaged to obtain the ensemble weights.

It allows H independent attention models to compute the attention weights with independent parameters in parallel. The attention weights are averaged to obtain the ensemble weights, as follows:

$$(w_{sk}^*, w_k^*) = \frac{1}{H} \left(\sum_{h=1}^{H} w_{sk}^h, \sum_{h=1}^{H} w_k^h \right), \ k = 1, \ldots, K \,, \tag{11}$$

where H is the number of attention models and w_{sk}^h represents the attention weights independently computed by the h-th attention model SPA^h. Then the combine features can be as follows:

$$\tilde{\boldsymbol{f}}_k = (w_{sk}^*, w_k^*) \cdot (\boldsymbol{f}_s, \boldsymbol{f}_k)^{\mathrm{T}}, \ k = 1, 2, \ldots, K. \tag{12}$$

E. Discussions

1) Differences from the Previous Works: MSJF is an approach to jointly forecast the time series of multiple related stocks based on DMTL, while most previous works only focus on a single stock. Moreover, most of the previous works in DMTL easily put all latent features together, but we aim to combine them with more useful knowledge. Since we mainly focus on financial data, so inspired by the idea of CAPM and Attention, we propose a new method, SPA, to learn the optimized combination of all latent features.

IV. Experiments

A. Dataset

1) Stock data of the Big Four banks in China: In this paper, we focus on forecasting stock time series, so we choose the stock daily trading data of the Big Four banks in China. The details of the dataset are presented in Table I.

These four stocks come from the Chinese banking industry, which are the most representative stocks in this industry. Each stock has 10 time series, including opening prices, closing prices, highest prices, lowest prices, trading volumes and so on.

There are four forecasting tasks that predict the excess returns of the next day for these four stocks.

TABLE I
DATASETS USED IN THE EXPERIMENTS. **TASKS** MEANS THE FORECASTING TASKS ASSIGNED IN THE DATASET.

Dataset	Description of the dataset		
	Period	Samples	Tasks
Banking stocks	2010-10 to 2018-08	1937	ICBC, ABC, BOC, CCB
Security stocks	2010-02 to 2018-08	2122	CITIC, CM, HAI, GF, HUA, EB
Shipping stocks	2010-01 to 2018-08	2143	CSSC, CSSC-TECH, HAIXUN, RUITE, HIGHLANDER, JIANGLONG, ASAC, BESTWAY

(a) Stock prices of the Big Four banks in China.

(b) Stock prices of six securities in China.

Fig. 4. Stock prices.

2) Stock data of six securities in China: Besides the stocks from the banking industry, we also choose the stock data of six securities in China. Similar to the Big Four banks in China, they are representative in the Chinese securities industry. The details are similar to the banking stock dataset, also presented in Table I.

There are 6 forecasting tasks that predict the excess returns of the next day for 6 different stocks.

3) Stock data of eight shipping stocks in China: In addition, we select 8 stocks from the industry of shipping in the Chinese market. There exist trading connections among these companies, so the trading time series of them are correlated. The input time series features are the same as the other datasets.

There are 8 forecasting tasks that respectively predict the temperature and humidity of 8 rooms in the house.

B. Baselines

We compare the proposed method with the following baseline methods, including linear methods and deep learning based methods:

- Moving Average model (MA): This is a classic linear method for time series forecasting, widely used in economics. So it serves as a baseline for comparison.
- Auto-Regressive Integrated Moving Average model (ARIMA): This is another classic linear method for time series forecasting, also serves as a baseline.
- Single-task Learning (ST): This serves as a baseline without benefits of multi-task learning. Each single-task model predicts one forecasting task, not sharing the information of other related tasks.
- Fully-connected Layers (FC): Many existing works treat related forecasting tasks as a multivariate time series forecasting problem and process it with a fully-connected neural network layer, so this method serves as a baseline.
- Fully-shared and Single-task (FSST): It also serves as a baseline without benefits of MTL, using the shared information of all tasks but still single-task.
- Fully-shared and Multi-task (FSMT): It serves as a baseline using only the shared information to forecast, similar to the previous works we mentioned, which can prove the benefits of our multi-model architecture.
- Private-shared MTL (PS-MTL): As our final baseline, we compare to a variant method of [37]. The original method builds multiple private encoding models and there is a shared embedding layer learning the shared representations of all private latent features, different from ours. So their method is adapted to this problem and serves as a private-shared MTL baseline.

C. Implement settings

For MA and ARIMA, the parameter settings are chosen through grid search with the best Bayesian information criterion (BIC) [40]. And their models are implemented with the APIs of *statsmodels* [41], a python package.

All deep learning based methods are implemented using *Tensorflow* [42]. And they use Long Short Term Memory Recurrent Neural Networks (LSTM-RNN) [43], [44] as their encoding models, fully-connected layers as their forecasting and attention models. The objective is MSE, and optimized by Adam [45] with Gradient Clipping [46]. The hyper-parameter settings and model details are presented in supplement pages.

D. Evaluation metrics

The loss functions are based on Mean Square Error (MSE), and three other metrics are used to evaluate the forecasting performance. Suppose that $\boldsymbol{y} = [y_{(1)}, y_{(2)}, \ldots, y_{(N)}]$ represents the ground truth, while $\hat{\boldsymbol{y}} = [\hat{y}_{(1)}, \hat{y}_{(2)}, \ldots, \hat{y}_{(N)}]$ represents the predicted values, and N denotes the number of samples, these metrics can be described as follows:

TABLE II

OVERALL PERFORMANCE COMPARISON. THE FORECASTING PERFORMANCES OF THE PROPOSED METHOD ON THREE DATASETS ARE MEASURED BY 4 METRICS, MSE, MAPE, MAE, AND MARE. OUR METHODS ARE MSJF, SPA-MSJF, AND MH-SPA-MSJF, WHILE THE OTHERS ARE BASELINE METHODS. THE EXPERIMENTS FOR EACH METHOD ARE RANDOMLY PERFORMED FIVE TIMES, AND THE RESULTS BELOW ARE THE AVERAGE TESTING VALUES OF ALL TASKS IN FIVE EXPERIMENTS. THE MSE VALUES ARE PRESENTED IN THE FORM OF $mean(\pm std)$.

	Banking stocks				Security stocks				Shipping stocks			
Method	MSE	MAPE	MAE	MARE	MSE	MAPE	MAE	MARE	MSE	MAPE	MAE	MARE
MA	3.78e-4	3.4003	0.0186	2.0319	6.78e-4	2.2511	0.6449	0.9038	4.44e-4	3.2312	0.3117	0.8305
ARIMA	3.61e-4	2.3114	0.0184	2.0088	4.54e-4	2.9347	0.4911	0.6879	3.72e-4	3.0091	0.3094	0.8105
ST	3.81e-4 ($\pm$2e-5)	3.3011	0.0119	1.2911	1.16e-4 ($\pm$1e-5)	1.3344	0.2649	0.3717	3.44e-4 ($\pm$3e-5)	3.0014	0.2901	0.7942
FC	2.87e-4 ($\pm$2e-5)	2.5078	0.0109	1.1872	1.16e-4 ($\pm$1e-5)	1.2128	0.2541	0.3569	3.25e-4 ($\pm$4e-5)	2.9442	0.2801	0.7245
FSST	2.66e-4 ($\pm$2e-5)	3.0274	0.0108	1.1738	1.09e-4 ($\pm$2e-5)	1.3175	0.2354	0.3300	2.44e-4 ($\pm$3e-5)	2.6617	0.2733	0.7642
FSMT	2.70e-4 ($\pm$2e-5)	3.1601	0.0111	1.1968	1.12e-4 ($\pm$2e-5)	1.1963	0.2351	0.3298	2.31e-4 ($\pm$1e-5)	2.3364	0.2513	0.7244
PSMTL	2.78e-4 ($\pm$2e-5)	2.5901	0.0109	1.1869	1.07e-4 ($\pm$3e-5)	1.2301	0.2358	0.3317	1.94e-4 ($\pm$4e-5)	2.1943	0.2756	0.7196
MJSF	2.51e-4 ($\pm$2e-5)	2.6485	0.0107	1.1621	1.04e-4 ($\pm$2e-5)	1.1303	0.2265	0.3177	1.72e-4 ($\pm$5e-5)	1.9341	0.1525	0.5465
SPA-MSJF	2.24e-4 ($\pm$1e-5)	2.4662	0.0103	1.1146	**1.01e-4** ($\pm$1e-5)	**1.1255**	0.2266	0.3178	**1.21e-4** ($\pm$6e-5)	1.5627	**0.0964**	**0.3427**
MH-SPA-MSJF	**2.03e-4** ($\pm$4e-6)	**2.1331**	**0.0097**	**1.0541**	1.02e-4 ($\pm$1e-5)	1.1504	**0.2232**	**0.3132**	1.32e-4 ($\pm$6e-5)	**1.1651**	0.1214	0.3544

TABLE III

PERFORMANCE COMPARISON ON INDIVIDUAL TASKS IN THE BANKING STOCK DATASET.

Task	ST	MSJF	SPA-MSJF	MH-SPA-MSJF
ICBC	3.91e-4	2.53e-4	2.34e-4	**2.05e-4**
ABC	2.36e-4	2.25e-4	1.99e-4	**1.83e-4**
CCB	5.38e-4	2.72e-4	2.28e-4	**2.21e-4**
BOC	3.61e-4	2.55e-4	2.34e-4	**2.04e-4**

TABLE IV

PERFORMANCE COMPARISON ON INDIVIDUAL TASKS IN THE SECURITY STOCK DATASET.

Task	ST	MSJF	SPA-MSJF	MH-SPA-MSJF
CITIC	1.173e-4	1.117e-4	**1.054e-4**	1.063e-4
CM	1.181e-4	1.053e-4	**1.001e-4**	1.052e-4
HAI	1.107e-4	0.976e-4	**0.962e-4**	0.965e-4
GF	1.156e-4	1.124e-4	**1.077e-4**	1.112e-4
HUA	1.182e-4	1.113e-4	**1.026e-4**	1.047e-4
EB	1.077e-4	1.012e-4	**0.998e-4**	1.004e-4

- Mean Absolute Percentage Error (MAPE):

$$\mathrm{MAPE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{i}^{N}\left|\frac{y_{(i)} - \hat{y}_{(i)}}{y_{(i)}}\right|, \tag{13}$$

- Mean Absolute Error (MAE):

$$\mathrm{MAE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{i}^{N}\left|y_{(i)} - \hat{y}_{(i)}\right|, \tag{14}$$

- Mean Absolute Relative Error (MARE):

$$\mathrm{MARE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\sum_{i}^{N}\left|y_{(i)} - \hat{y}_{(i)}\right|}{\sum_{i}^{N}\left|y_{(i)}\right|}. \tag{15}$$

E. Results and Analysis

1) Overall Performance Comparison: The overall comparison experiment results are shown in Table II. From these results, We have the following observations: 1) Both MSJF and SPA-MSJF can outperform the baseline methods on all datasets. This indicates the effectiveness of the proposed methods; 2) SPA-MSJF is better than MSJF. This demonstrates the proposed SPA model can indeed further improve the performance of MSJF.

TABLE V

PERFORMANCE COMPARISON ON INDIVIDUAL TASKS IN THE SHIPPING STOCK DATASET.

Task	ST	MSJF	SPA-MSJF	MH-SPA-MSJF
CSSC	3.217e-4	1.608e-4	1.194e-4	**1.126e-4**
CSSC-TECH	3.174e-4	1.673e-4	**1.119e-4**	1.258e-4
HAIXUN	2.918e-4	1.776e-4	1.242e-4	**1.217e-4**
RUITE	2.946e-4	1.693e-4	**1.172e-4**	1.189e-4
HIGHLANDER	2.992e-4	1.926e-4	1.225e-4	**1.147e-4**
JIANGLONG	3.319e-4	1.895e-4	**1.155e-4**	1.265e-4
ASAC	3.376e-4	1.678e-4	**1.159e-4**	1.305e-4
BESTWAY	3.005e-4	1.533e-4	1.151e-4	**1.141e-4**

2) Effects of Multi-series Jointly Forecasting: To show the effects of MSJF, we use the experimental results in Table II, III, IV and V. Without the benefits of SPA, 1) in Table II, MSJF outperforms single-task (ST) and fully-shared & single-task (FSST) baselines. And it outperforms ST on each task in all datasets, as shown in Table III, IV and V. These suggest the effectiveness of MSJF; 2) MSJF performs better than fully-shared & multi-task (FSMT) and private-shared MTL (PS-MTL) baselines. This suggests the effectiveness of the multi-model architecture in MSJF.

3) Analysis on Shared-private Attention: On the basis of MSJF, we propose SPA to learn the optimized combination of shared and task-private latent features. In Table II, SPA-MSJF outperforms MSJF on the average test MSE in all datasets. And in Table III, IV, V, SPA-MSJF outperforms MSJF on 18 tasks (totally 18 tasks). These results demonstrate the effectiveness of SPA.

We also provide a visualization of combination weights learned by SPA, shown in Figure 6. From the visualization, 1) we can find the shared weights are larger than the private weights in almost all test data of stock datasets. It means the shared information plays an important role in stock forecasting, which is similar to the conclusion of CAPM. This

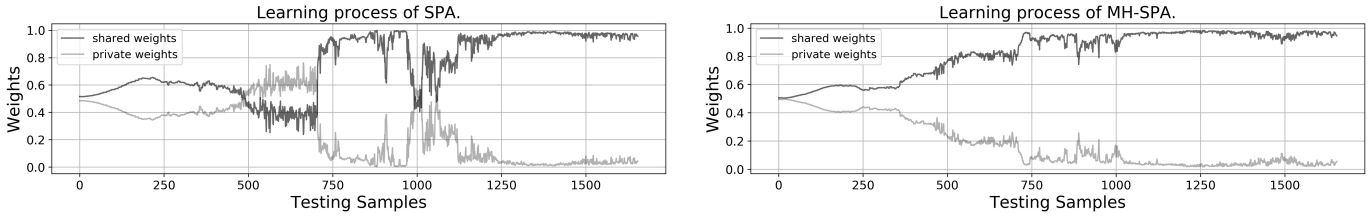

Fig. 5. The attention weights learned by Shared-private Attention (SPA) and Multi-head Shared-private Attention (MH-SPA) on the stock dataset. We use sliding training and testing on the stock dataset, so these figures show the change of w_{sk} and w_k during the training process, indicating that the learning process is stabilized by the multi-head mechanism.

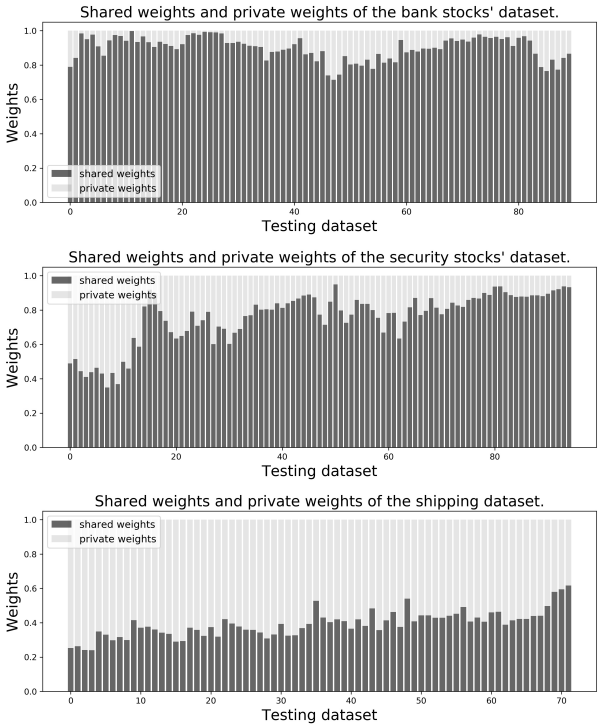

Fig. 6. The attention weights learned by Shared-private Attention (SPA). To more clearly show the results, all the testing samples are evenly divided into several subsets. The weights shown are the average of the samples in each set.

result indicates the SPA model can indeed leverage the idea of CAPM to improve the forecasting performance; 2) As for the result in the shipping stock dataset, we find a different pattern: the shared weights are almost the same as the private weights. However, from the result in Table V, SPA-MSJF is still better than MSJF on average. This shows SPA also can work on the non-stock data. These results also demonstrate the effectiveness of SPA.

4) Effects of the Multi-head Mechanism: In experiments, we find the learning process of SPA on the stock dataset is not stable, so we apply the multi-head mechanism and obtain Multi-head Shared-private Attention. We provide visualization for the learning processes of SPA and MH-SPA in Figure 5. We can find that, with the benefits of the multi-head mechanism, the learning process is stabilized. And in Table II, MH-SPA-MSJF outperforms SPA-MSJF on the stock dataset. These demonstrate the effectiveness of MH-SPA.

Experimental results on the stock time series datasets demonstrate the proposed methods, MSJF, SPA-MSJF and MH-SPA-MSJF, outperform the previous works, including linear methods, single-task methods and other DMTL based solutions. We separately analyze the effects of MSJF, SPA, and MH-SPA, using the results to prove they further improve the performance indeed. In addition, with the visualizations, we analyze the effectiveness of SPA and MH-SPA in further details.

V. Conclusion

In this paper, we propose a jointly forecasting approach, MSJF, to process the time series of multiple related stocks based on DMTL, which can use the connections among stocks to improve the forecasting performance. Moreover, in order to combine the shared and task-private information more accurately, we propose an attention method, SPA, to learn the optimized combination of them inspired by the idea of CAPM. We demonstrate our method on various financial datasets, and it outperforms the classic methods and other MTL based methods. In the future works, we would like to further improve SPA's ability of combining latent features. And for DMTL, we would like to build hierarchical models to extract the shared information from all tasks more efficiently.

VI. Acknowledgements

This work is supported by the National Natural Science Foundation of China (GrantNos.: 61673025, 61375119), and the National Key R&D Program of China (Grant Nos.: 2018AAA0100300, 2018AAA0102301), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China (Grant No. 2015CB352302).

References

[1] L. Laloux, P. Cizeau, M. Potters, and J.-P. Bouchaud, "Random matrix theory and financial correlations," *International Journal of Theoretical and Applied Finance*, vol. 3, no. 03, pp. 391–397, 2000.

[2] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," *Quantitative Finance*, vol. 1, no. 2, pp. 223–236, 2001.

[3] B. G. Malkiel and E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[4] J. D. Hamilton, *Time series analysis*. Princeton university press Princeton, NJ, 1994, vol. 2.

[5] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka, "Forecasting the behavior of multivariate time series using neural networks," *Neural networks*, vol. 5, no. 6, pp. 961–970, 1992.

[6] P.-F. Pai and C.-S. Lin, "A hybrid arima and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.

[7] W. F. Sharpe, "Capital asset prices: A theory of market equilibrium under conditions of risk," *The journal of finance*, vol. 19, no. 3, pp. 425–442, 1964.

[8] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[9] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE transactions on image processing*, vol. 25, no. 6, pp. 2529–2541, 2016.

[10] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in neural information processing systems*, 2014, pp. 1988–1996.

[11] H. Harutyunyan, H. Khachatrian, D. C. Kale, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *CoRR*, vol. abs/1703.07771, 2017. [Online]. Available: http://arxiv.org/abs/1703.07771

[12] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *International Conference on Knowledge Discovery and Data Mining,(KDD)*, 2018.

[13] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *CoRR*, vol. abs/1709.01507, 2017. [Online]. Available: http://arxiv.org/abs/1709.01507

[14] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia, "Multi-task cnn model for attribute prediction," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1949–1959, 2015.

[15] G. E. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *Journal of the American statistical Association*, vol. 65, no. 332, pp. 1509–1526, 1970.

[16] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[17] Y. Hwang, A. Tong, and J. Choi, "Automatic construction of nonparametric relational regression models for multiple time series," in *International Conference on Machine Learning*, 2016, pp. 3030–3039.

[18] J. Cunningham, Z. Ghahramani, and C. Rasmussen, "Gaussian processes for time-marked time-series data," in *Artificial Intelligence and Statistics*, 2012, pp. 255–263.

[19] A. G. Wilson and Z. Ghahramani, "Copula processes," in *Advances in Neural Information Processing Systems*, 2010, pp. 2460–2468.

[20] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[23] R. Józefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *CoRR*, vol. abs/1602.02410, 2016. [Online]. Available: http://arxiv.org/abs/1602.02410

[24] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.

[25] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition." in *Ijcai*, vol. 15, 2015, pp. 3995–4001.

[26] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang *et al.*, "Traffic flow prediction with big data: A deep learning approach." *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[27] X. Zhang and Y. Tan, "Deep stock ranker: A LSTM neural network model for stock selection," in *Data Mining and Big Data - Third International Conference, DMBD 2018, Shanghai, China, June 17-22, 2018, Proceedings*, ser. Lecture Notes in Computer Science, Y. Tan, Y. Shi, and Q. Tang, Eds., vol. 10943. Springer, 2018, pp. 614–623. [Online]. Available: https://doi.org/10.1007/978-3-319-93803-5_58

[28] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. Wooldridge, Eds. AAAI Press, 2015, pp. 2327–2333. [Online]. Available: http://ijcai.org/Abstract/15/329

[29] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning in finance," *CoRR*, vol. abs/1602.06561, 2016. [Online]. Available: http://arxiv.org/abs/1602.06561

[30] D. Neil, M. Pfeiffer, and S.-C. Liu, "Phased lstm: Accelerating recurrent network training for long or event-based sequences," in *Advances in Neural Information Processing Systems*, 2016, pp. 3882–3890.

[31] M. Binkowski, G. Marti, and P. Donnat, "Autoregressive convolutional neural networks for asynchronous time series," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. JMLR Workshop and Conference Proceedings, J. G. Dy and A. Krause, Eds., vol. 80. JMLR.org, 2018, pp. 579–588. [Online]. Available: http://proceedings.mlr.press/v80/binkowski18a.html

[32] S. Ruder, "An overview of multi-task learning in deep neural networks," *CoRR*, vol. abs/1706.05098, 2017. [Online]. Available: http://arxiv.org/abs/1706.05098

[33] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[34] R. Dürichen, M. A. Pimentel, L. Clifton, A. Schweikard, and D. A. Clifton, "Multitask gaussian processes for multivariate physiological time-series analysis," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 1, pp. 314–322, 2015.

[35] A. Jung, "Learning the conditional independence structure of stationary time series: A multitask learning approach," *IEEE Transactions on Signal Processing*, vol. 63, no. 21, pp. 5677–5690, 2015.

[36] Y. Liu, Y. Zheng, Y. Liang, S. Liu, and D. S. Rosenblum, "Urban water quality prediction based on multi-task multi-view learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, S. Kambhampati, Ed. IJCAI/AAAI Press, 2016, pp. 2576–2581. [Online]. Available: http://www.ijcai.org/Abstract/16/366

[37] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.

[38] M. C. Jensen, "The performance of mutual funds in the period 1945–1964," *The Journal of finance*, vol. 23, no. 2, pp. 389–416, 1968.

[39] 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*, 2017, pp. 5998–6008.

[40] S. I. Vrieze, "Model selection and psychological theory: a discussion of the differences between the akaike information criterion (aic) and the bayesian information criterion (bic)." *Psychological methods*, vol. 17, no. 2, p. 228, 2012.

[41] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, vol. 57. SciPy society Austin, 2010, p. 61.

[42] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[44] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4580–4584.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[46] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.