第51卷第3期 中国科学技术大学学报 Vol. 51, No. 3
2021年3月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Mar. 2021

Received: 2021-02-18; Revised: 2021-03-24 doi:10.52396/JUST-2021-0052

# An end-to-end multitask method with two targets for high-frequency price movement prediction

MA Yulian[1,2], CUI Wenquan[1,2*]

1. International Institute of Finance, University of Science and Technology of China, Hefei 230601 ,China;
2. School of Management, University of Science and of Technology of China, Hefei 230026,China
*Corresponding author. E-mail: wqcui@ustc.edu.cn

**Abstract**: High-frequency price movement prediction is to predict the direction(e. g. up, unchanged or down) of the price change in short time(e. g. one minute). It is challenging to use historical high-frequency transaction data to predict price movement because their relation is noisy, nonlinear and complex. We propose an end-to-end multitask method with two targets to improve high-frequency price movement prediction. Specifically, the proposed method introduces an auxiliary target(high-frequency rate of price change), which is highly related with the main target(high-frequency price movement) and is useful to improve the high-frequency price movement prediction. Moreover, each task has a feature extractor based on recurrent neural network and convolutional neural network to learn the noisy, nonlinear and complex temporal-spatial relation between the historical transaction data and the two targets. Besides, the shared parts and task-specific parts of each task are separated explicitly to alleviate the potential negative transfer caused by the multitask method. Moreover, a gradient balancing approach is adopted to use the close relation between two targets to filter the temporal-spatial dependency learned from the inconsistent noise and retain the dependency learned from the consistent true information to improve the high-frequency price movement prediction. The experimental results on real-world datasets show that the proposed method manages to utilize the highly related auxiliary target to help the feature extractor of the main task to learn the temporal-spatial dependency with more generalization to improve high-frequency price movement prediction. Moreover, the auxiliary target(high-frequency rate of the price change) not only improves the generalization of overall temporal-spatial dependency learned by the whole feature extractor but also improve temporal-spatial dependency learned by the different parts of the feature extractor.

**Keywords**: multitask learning; fine-grained auxiliary target; feature extraction; sharing method; negative transfer; high-frequency price movement prediction

**CLC number**: O212; O234; TP181 **Document code**: A

## 1 Introduction

The asset price movement prediction is an open and interesting problem. First of all, there are two contradictory perspectives on its predictability and both of them achieve the Nobel Memorial Prize in Economic Sciences. The one is the efficient market hypothesis (EMH) of Fama, stating that it is impossible to predict the asset price movement[1] and the other one is the behavioural economics of Thaler, implying that the asset price movement is somewhat predictable[2]. Compared with the EMH, the behavioural economics is based on the bounded rationality rather than the rational agent assumption, so it is more realistic in most cases[3]. For example, the famous monthly effect[4] and the small firm effect[5] are included in the behavioural economics, which show that the asset price movement can be predicted to some extent.

As a result, on the basis of the behavioural economics, a lot of solutions are proposed to predict the price movement[6-9]. On the whole, these solutions try to improve the price movement prediction either by using more diverse unstructured input data or by constructing more elaborate classifiers. On the one hand, besides the traditional structured data (e. g. technical indicators[10-12]), social media information like tweets and blogs[13-16] and company-related financial news[14,16-18] are utilized to predict price

**Citation**: MA Yulian, CUI Wenquan. An end-to-end multitask method with two targets for high-frequency price movement prediction. J. Univ. Sci. Tech. China, 2021, 51(3): 246-258.

movement. On the other hand, more elaborate classifiers based on deep learning (e. g. the long short-term memory (LSTM)[19,20], convolutional neural network[20,21]), the reinforcement learning (e. g. deep Q-learning[22]) and the generative adversarial training[12] are proposed to improve the price movement prediction.

Although these solutions manage to improve the price movement prediction, most of them are single task method, which just construct the classification task by directly taking sign value of price change as the sole target and pay little attention to the related and more fine-grained rate of price change (containing the information about the magnitude of price change except the sign value). However, it has been empirically shown that the task with a fine-grained target probably synergizes with the task with a coarse-grained target[13], which indicates that the task with more fine-grained target can probably improve performance of the classification task with the coarse-grained price movement target. Besides, it can be proven that given the input variables $x$, the two-target multitask learning method jointly modeling the distributions of two related targets ($y$ and $r$) $P(y|r,x)$ and $P(r|y,x)$ will get the more certain optimal distribution $P(y|x)$ than the single task method, which models the distribution $P(y|x)$ without using the variable $r$ so that the proposed two-target multitask method jointly modeling the distributions of two highly related targets (the high-frequency price movement and rate of price change)[24-27] can get the more certain distribution about high-frequency price movement to further improve its prediction. Specifically, we give the following proof. For the crossentropy loss function for classification task, the optimal distribution is the target distribution and the optimal value is the entropy of target distribution so that the optimal value of the two-target multitask learning method is the entropy

$$H(P(y \mid r,x),q) = -\sum_{i=1}^{3} P(y_i \mid r,x)\log q(y_i \mid r,x)$$

and the optimal value of the single task method is the entropy

$$H(P(y \mid x),p) = -\sum_{i=1}^{3} P(y_i \mid x)\log p(y_i \mid x).$$

Because of $H(y \mid r,x) \leqslant H(y \mid x)$, the two-target multitask learning method will get a more certain optimal distribution for high-frequency price movement and get more certain prediction than the single task method. Furthermore, the difference $H(y|x)-H(y|r,x)$ is increasing with the relation between the two targets ($y$ and $r$) and when $y$ and $r$ is independent, the difference reaches 0 and when $y$ is the same as $r$ (totally related), the difference reaches the max value $H(y|x)$.

Given an extra variable (e. g. rate of price change), it is natural to take it as an the input variable[28,29], but the complex and poor linear correlation between the high-frequency rate of price change and price movement[30] and the high noise of the rate of price change[31,32] will probably deteriorate the capability of the back propagation to learn to use the rate of price change to predict price movement[28,33-35]. Different from directly taking the rate of price change as input, the proposed two-target method takes it as the auxiliary target to leverage its relation with the price movement and can bias the hidden layer to encode useful representation for price movement prediction even when the rate of price change is noisy and the linear correlation between them is weak[28] to get better price movement prediction[36]. Besides, when it comes to choose the concrete variable on behalf of the rate of price change, the two-target multitask method enjoys more flexibility than the method directly taking the variable as input[37] because the two-target multitask method can not only choose the variable accessible before the prediction time but also the variable accessible after prediction time, while the direct input method can only choose the variable accessible before the prediction time[38].

However, there are two challenges in designing the two-target multitask method: ① how to design its feature extractor to learn the noisy, nonlinear and complex temporal-spatial dependency; ② how to choose its sharing method to use the close relation between the two targets.

On one hand, it is an open problem to learn the temporal-spatial dependency between the historical high-frequency transaction data and the high-frequency price movement because of the intrinsic complexity, nonlinearity, dynamics and high noise of financial data[30,39-41]. As a result, we design the feature extractor from the global perspective rather than pursuing the optimal feature extractor applicable for all cases regardless of the other parts of the proposed method. Considering that the purpose of the method is to utilize the auxiliary target (high-frequency rate of price change) to learn the temporal-spatial dependency of more generalization of the main task to improve high-frequency price movement prediction, the feature extractor is supposed to be able to learn diverse temporal-spatial dependency for further processing of other parts of the method[42] so that the feature extractor is designed to be a combination of two different modules based on the convolutional neural network and recurrent neural network, which are the common core blocks for the feature extraction of price movement prediction task[10,19,21].

On the other hand, it is hard to choose the best sharing method suitable for all multitask method because the relation between tasks is variable and even unclear in most cases[38]. Considering that the feature extractor of the proposed method is designed to be complex to learn diverse temporal-spatial dependency, it is likely to learn from the noise rather than the true value, which is detrimental to the generalization according to the Occam’s Razor[43] so that the sharing method is expected to be able to deal with the noise[44,45] to improve the prediction. Finally, a newly gradient balancing approach (GradDrop)[46] is adopted as the sharing method, which merely updates the selected parameters for each iteration.

In summary, this paper has the following contributions:

(Ⅰ) Different from the most existing research on the high-frequency price movement prediction, which pays little attention to the close relation between the coarse-grained direction and the fine-grained rate of the price change, this paper proposes an end-to-end multitask method with two targets to incorporate the close relation to improve the high-frequency price movement prediction.

(Ⅱ) A feature extractor is designed to learn diverse temporal-spatial dependency from the historical transaction data for two targets for further processing of the other parts of the method.

(Ⅲ) A sharing method is adopted to use the close relation between two targets to filter the temporal-spatial dependency learned from the inconsistent noise and retain the dependency learned from the consistent true information to improve the high-frequency price movement prediction.

## 2 Methodology

The structure of the proposed multitask method is shown in Figure 1, which consists of three parts, the main task to predict the high-frequency price movement (see Section 2.1), the auxiliary task to enhance feature extraction of main task (see Section 2.2) and the sharing method to communicate information between tasks (see Section 2.3). Moreover, each task contains a feature extractor to learn diverse temporal-spatial dependency for further processing of other parts of the method and a prediction layer to fuse the dependency to get the specific prediction target. We will introduce them in detail step by step.

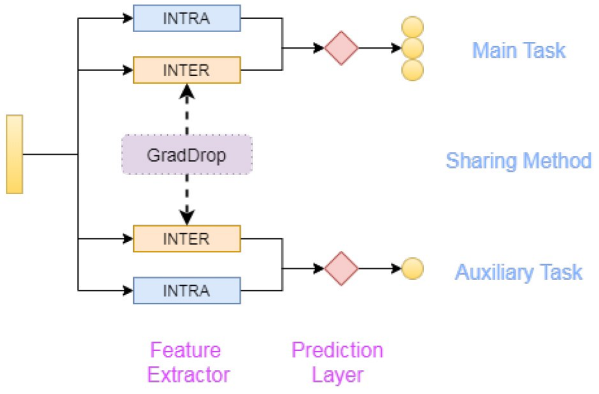

**Figure 1.** Structure of the proposed method.

### 2.1 The main task

The main task is a classification task to predict the future price movement (up, unchanged or down) within a short time (e.g. one minute) with the raw historical high-frequency (e.g. one minute) transaction data. The input transaction data is a multivariate time series for $T$ time steps, denoted by $X=\{X^1, X^2, \cdots, X^T\}$, with each $X^t \in \mathbb{R}^p$ containing $p$ elements, such as open price, highest price, lowest price, close price, volume and position. The prediction target is the price movement, sign value of difference between close price and open price, denoted by $Y=\{y^1, y^2, \cdots, y^T\}$,

$$y^t = \begin{cases} 1, & \text{close\_price}^t > \text{open\_price}^t \\ 0, & \text{close\_price}^t = \text{open\_price}^t \\ -1, & \text{close\_price}^t < \text{open\_price}^t \end{cases} \qquad (1)$$

where close_price$^t$ and open_price$^t$ are elements of $X^t$. The goal of the main task is to learn a classifier $P(y^{t+1} \mid X^t, X^{t-1}, \cdots, X^{t-k+1})$, $t=k, k+1, \cdots, T-1$ and the $k<T$ is a positive integer representing the rolling windows width.

#### 2.1.1 The feature extractor

The feature extractor is the parallel combination of an INTRA module and an INTER module to learn diverse temporal-spatial dependency for further processing of other parts of the method[42]. Specifically, the INTRA module is based on the recurrent neural network to learn the dynamic sequential temporal-spatial dependency (see Section 2.1.1.1) and the INTER module is based on the convolutional neural network to learn static accumulative temporal-spatial dependency (see Section 2.1.1.2).

On the one hand, the recurrent neural network and the convolutional neural network are of great learning capacity[47,48] to extract diverse temporal-spatial dependency[42] from the limited transaction input data because of their elaborate architectures (e.g. sequential calculation, shared weights and local connection).

On the other hand, because the structures of the convolutional neural network and recurrent neural network are greatly different[49,50], the combination of them can probably extract more diverse temporal-spatial dependency than the single network[42], which can increase the possibility to provide the temporal-spatial dependency of generalization for further processing of other parts of the proposed method to improve the high-frequency price movement prediction[13-16].

#### 2.1.1.1 The INTRA module

The INTRA module is designed to learn the dynamic sequential temporal-spatial dependency, which is implemented by the recurrent neural network with LSTM unit. A common LSTM unit contains forget gate, input gate and output gate to control the information flow. Specifically, given the input $X^t$, $X^{t-1},\cdots,X^{t-k+1}$ and the LSTM unit's hidden states size $d$, the output $h^t=\mathrm{INTRA}(X^t, X^{t-1},\cdots,X^{t-k+1})$ is calculated by equations

$$\begin{cases} i^t = \sigma(W_{ii}X^t + b_{ii} + W_{hi}h^{t-1} + b_{hi}) \\ f^t = \sigma(W_{if}X^t + b_{if} + W_{hf}h^{t-1} + b_{hf}) \\ o^t = \sigma(W_{io}X^t + b_{io} + W_{ho}h^{t-1} + b_{ho}) \\ g^t = \tanh(W_{ig}X^t + b_{ig} + W_{hg}h^{t-1} + b_{hg}) \\ c^t = i^t \odot g^t + f^t \odot c^{t-1} \\ h^t = o^t \odot \tanh(c^t) \end{cases} \tag{2}$$

where $\sigma(x)=\dfrac{1}{1+\mathrm{e}^{-x}}$ and $\tanh(x)=\dfrac{\mathrm{e}^x-\mathrm{e}^{-x}}{\mathrm{e}^x+\mathrm{e}^{-x}}$; $i^t$, $f^t$, $o^t\in(0,1)^d$ are the outputs of input gate, forget gate and output gate respectively; $g^t$, $c^t$, $h^t\in(-1,1)^d$ represent the information flow; $W_{\mathrm{yintra}}=\{W_{ii}, W_{hi}, W_{if}, W_{hf}, W_{io}, W_{ho}, W_{ig}, W_{hg}, b_{ii}, b_{hi}, b_{if}, b_{hf}, b_{io}, b_{ho}, b_{ig}, b_{hg}\}$ are the learnable parameters with $W_{ii}$, $W_{if}$, $W_{io}$, $W_{ig}\in\mathbb{R}^{d\times p}$, $W_{hi}$, $W_{hf}$, $W_{ho}$, $W_{hg}\in\mathbb{R}^{d\times d}$, $b_{ii}$, $b_{hi}$, $b_{if}$, $b_{hf}$, $b_{io}$, $b_{ho}$, $b_{ig}$, $b_{hg}\in\mathbb{R}$.

Due to the ranges of outputs of three gates are (0, 1), they can be interpreted as the ratios for updating. More specifically, the input gate $i^t$ is to protect the stored memory contents from perturbation by unrelated inputs and the forget gate $f^t$ as well as the output gate $o^t$ are used to protect other units from perturbation by presently unrelated stored memory contents[50]. As a result, the devotion of each input vector at different time to the final feature is decided in a data-driven way so that the devotion is totally temporally variable, which means that the feature is dynamic. Besides, with the sequential calculation of the features, the INTRA module can adaptively learn the global sequential temporal patterns. In summary, the INTRA module can learn the global dynamic sequential patterns.

#### 2.1.1.2 The INTER module

The INTER module is designed to learn the static temporal-spatial dependency, implemented by convolutional neural network. The structure of INTER is shown in Figure 2, the parallel connection of three one-layer CNNs, a temporal aggregation layer and the concatenation layer, where the CNNs are used to learn the local temporal-spatial features of temporal invariance, the temporal aggregation layer is to obtain the accumulative temporal effects of the local patterns to grasp the global temporal-spatial patterns and the concatenation layer is to concatenate the extracted features from three CNNs to an integrate feature vector.

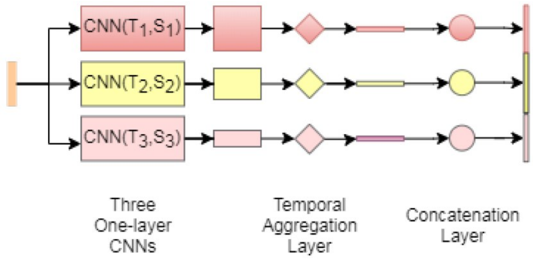

**Figure 2.** Structure of the INTER module.

Specifically, the sizes of the filters of the three types of CNNs are different to extract the short-term, medium-term and long-term patterns. Given the input $\{X^t, X^{t-1},\cdots,X^{t-k+1}\}$, the output feature vector is denoted by $v^t=\mathrm{INTER}(X^t, X^{t-1},\cdots,X^{t-k+1})$.

Firstly, given that the $i^{\mathrm{th}}$ CNN has $F$ filters with size $(T_i, p)$, where $T_i$ is the time dimension and $p$ is the variable dimension, with $0<T_1<T_2<T_3<k$, $i=1,2,3$, the outputs of the three CNNs, denoted by $\{u_i^t\}\in(-1,1)^{F\times(k-T_i+1)}$, $i=1,2,3$, are calculated by

$$u_{iac}{}^{t} = \tanh(\mathbf{1}_p'(W^{ia}\odot[X^{c+t-k},\cdots,X^{c+t-k+T_i-1}])\mathbf{1}_{T_i}+b^{ia}) \tag{3}$$

where $a=1, 2,\cdots, F$ and $c=1, 2,\cdots, k-T_i+1$; the elements of vectors $\mathbf{1}_p$ and $\mathbf{1}_{T_i}$ are all one; $\{W^{ia}\in\mathbb{R}^{p\times T_i}$, $b^{ia}\in\mathbb{R}$, $i=1, 2, 3$, $a=1, 2,\cdots, F\}$ are the set of learnable parameters, denoted by $W_{\mathrm{yinter}}$. Secondly, the $u_i^t$, $i=1,2,3$, are added along time dimension to get the temporal accumulative output, denoted by $r_i^t\in\mathbb{R}^F$, $i=1,2,3$,

$$r_i^t = u_i^t\mathbf{1}_{k-T_i+1} \tag{4}$$

where elements of $\mathbf{1}_{k-T_i+1}$ are all one. Thirdly, concatenate $r_t^1$, $r_t^2$, $r_t^3$ to get the one-dimensional output feature vector $v^t\in\mathbb{R}^{3F}$,

$$v^t = \begin{pmatrix} r_1^t \\ r_2^t \\ r_3^t \end{pmatrix} \tag{5}$$

Once $T_1$, $T_2$ and $T_3$ are given, the filter with size $(T_1,p)$ is designed to learn the short-term patterns, the filter with size $(T_2,p)$ is to learn the medium-term patterns and the filter of size $(T_3,p)$ is to learn the long-term patterns. Moreover, with the temporally local connection (with $0<T_i<k$, $i=1,2,3$) and the weights sharing of the hidden units between two adjacent layers, the features are temporally invariant (static). And with the temporal aggregation of the local temporal patterns, the INTER module can learn the global accumulative static patterns.

### 2.1.2 The prediction layer

The prediction layer is used to fuse the learned temporal-spatial dependency to predict the future price movement. Besides, due to the outputs of the main task are probabilities for the three classes(up, unchanged or down), the corresponding prediction layer is a fully connected layer with the softmax activation.

Specifically, with the input being the combination of the extracted feature vectors of the two modules $h_y^t$ and $v_y^t$, the output of the prediction layer $\hat{y}^t$ is calculated by

$$\hat{y}^t = \text{softmax}(W_{yo}[h^{t'}_y, v^{t'}_y] + b_{yo}) \qquad (6)$$

where $\hat{y}^t \in (0,1)^3$ with $\hat{y}_1^t + \hat{y}_2^t + \hat{y}_3^t = 1$ are the probabilities for up, unchanged and down price movement; $h_y^{t'}, v_y^{t'}$ are the transpose of $h_y^t, v_y^t$; $\{W^{yo} \in \mathbb{R}^{3\times(d+3F)}, b^{yo} \in \mathbb{R}\}$ are the set of learnable parameters, denoted by $W_{\text{ypred}}$.

### 2.1.3 The loss function

The main task to predict price movement is a three-class classification task so that the cross entropy is used as the loss function. Suppose the true observation of price movement for the training dataset is $y = \{y^{t_1}, y^{t_2}, \cdots, y^{t_N}\}$, $y^{t_n} \in \{-1, 0, 1\}$, $n = 1, 2, \cdots, N$ and its corresponding probabilities estimation is $\widehat{y} = \{\hat{y}^{t_1 - 1}, \hat{y}^{t_2 - 1}, \cdots, \hat{y}^{t_N - 1}\}$, $\hat{y}^{t_n - 1} \in (0, 1)^3$, $n = 1, 2, \cdots, N$, the training loss is calculated as equation

$$L(y, \widehat{y}) = -\frac{1}{N}\sum_{n=1}^{N} \log \hat{y}^{t_n - 1}_{1+y^t{}_n}. \qquad (7)$$

## 2.2 The auxiliary task

The auxiliary task is a regression task to improve the performance of the main task (high-frequency price movement prediction) by enhancing the learning of temporal-spatial dependency of the main task with the auxiliary target and the sharing method. Specifically, the input data and the structure of feature extractor of the auxiliary task are the same with the main task, but the prediction target is different.

### 2.2.1 The auxiliary target

Choosing the high-frequency rate of price change as the auxiliary target can probably enhance the learning of temporal-spatial dependency of the main task to improve the price movement prediction[23]. Denote the high-frequency rate of price change by $Z = \{z^1, z^2, \cdots, z^T\}$, $z^t \in \mathbb{R}$,

$$z^t = \frac{\text{close\_price}^t - \text{open\_price}^t}{\text{open\_price}^t + \epsilon} \qquad (8)$$

where close_price$^t$ and open_price$^t$ are the elements of $X^t$; the $\epsilon$ is a sufficient small positive number (e.g. 0.000001) compared with close price and open price to deal with the case when open price equals zero and cause little effect when open price is nonzero.

The rate of price change is more fine-grained than the price movement because it contains both the direction (the price movement) and the exact magnitude of price change, which is empirically able to enhance the learning of temporal-spatial dependency of the main task to improve the high-frequency price movement prediction[23]. The high-frequency rate of price change is feasible to be auxiliary target of the multitask method because it is related (see Table 1) with the main target (the high-frequency price movement) as $y^t = \frac{z^t}{|z^t|}$, which makes the auxiliary task related with the main task to probably improve the performance of the main task[38].

**Table 1.** Close relation between the price movement and rate of price change.

| Datasets | Correlation | Mutual information ratio(%) | |
|---|---|---|---|
| | | Mean | Median |
| HC | 0.78 | 95.72 | 97.63 |
| I | 0.88 | 95.43 | 97.50 |
| J | 0.76 | 95.87 | 97.69 |
| NFLX | 0.68 | 95.52 | 97.59 |
| AMZN | 0.66 | 95.31 | 97.42 |
| AAPL | 0.68 | 96.07 | 97.83 |

[Note] ①The HC, I, J, NFLX, AMZN and AAPL are names of different futures or stocks, more details in Section 3.
② The training, validation and testing are the partition of the raw data.

### 2.2.2 The feature extractor

The structure of feature extractor of auxiliary task is the same as that of the main task, a parallel connection of one INTRA module and one INTER module. Specifically, given input $\{X^t, X^{t-1}, \cdots, X^{t-k+1}\}$, the features of the INTRA module and the INTER module are $\{h_z^t\} = \text{INTRA}(X^t, X^{t-1}, \cdots, X^{t-k+1} | W_{\text{zintra}})$ and $v_z^t = \text{INTER}(X^t, X^{t-1}, \cdots, X^{t-k+1} | W_{\text{zinter}})$ respectively, where the $W_{\text{zintra}}$ and $W_{\text{zintra}}$ are the learnable parameters of the feature extractor of auxiliary task.

### 2.2.3 The prediction layer

Different from the probability output of main task, the output of auxiliary output is a real number so that the corresponding prediction layer is a fully connected layer without the softmax activation. Denote $\hat{z}^t$ the output of prediction layer of auxiliary task and then

$$\hat{z}^t = W_{zo}[h^{t'}_z, v^{t'}_z] + b_{zo} \qquad (9)$$

where $\hat{z}^t \in \mathbb{R}$ is the rate of price change prediction; $h_z^{t'}, v_z^{t'}$ are the transpose of $h_z^t, v_z^t$; $W_{\text{zpred}} = \{W^{zo} \in \mathbb{R}^{1\times(d+3F)}, b^{yo} \in \mathbb{R}\}$ are the learnable parameters.

### 2.2.4 The loss function

The auxiliary task is a regression task so that the mean squared error is utilized as the loss function. Suppose the true observation of rate of price change for the training dataset is $z = \{z^{t_1}, z^{t_2}, \cdots, z^{t_N}\}$, $z^{t_n} \in \mathbb{R}$, $n = 1, 2, \cdots, N$, and its corresponding estimation is $\hat{z} = \{\hat{z}^{t_1 - 1},$

$\hat{z}^{t_2-1}, \cdots, \hat{z}^{t_N-1}\}$, $\hat{z}^{t_n-1} \in \mathbb{R}$, $n=1, 2, \cdots, N$, the training loss is calculated by equation

$$L(z, \hat{z}) = \frac{1}{N} \sum_{n=1}^{N} (\hat{z}^{t_n-1} - z^{t_n})^2 \tag{10}$$

## 2.3 The sharing method

Through the sharing method, the main task is capable of communicating information with the auxiliary task so that the main task has the potential to incorporate the related information of the auxiliary task to learn temporal-spatial dependency of more generalization for high-frequency price movement prediction[51-54]. Moreover, the sharing method usually contains two parts: ① the sharing structure to decide the shared components and the task-specific components of different tasks; ② the sharing mechanism to decide concrete communication approach between the shared components.

### 2.3.1 The sharing structure

The sharing structure of the proposed method is that the INTER modules of the two tasks are used as the shared components and the INTRA modules are used as the task-specific components. The shared components and the task-specific components are explicitly separated to alleviate the potential negative transfer caused by the multitask learning method[38,55] to use the relation between tasks to improve the performance of the main task (high-frequency price movement prediction)[56]. Although the high-frequency rate of the price change chosen as the auxiliary target is related with the main target (the price movement), both the targets are highly noisy[41,57] so that the multitask learning probably suffers the negative transfer[58], which will impair the high-frequency price movement prediction. Besides, the feature extractor designed to extract diverse temporal-spatial dependency is so complex that it is inevitable to incur more noise[43-45], which will probably further aggravate the negative transfer to degrade the generalization of multitask learning[58]. As a result, explicitly separating the INTER modules and the INTRA modules is probably to alleviate the potential detrimental interference between common and task-specific knowledge, which is helpful to use the relevant information of the auxiliary task to improve high-frequency price movement prediction[56].

### 2.3.2 The sharing mechanism

A soft sharing mechanism is utilized to communicate information between tasks not only by filtering inconsistent gradients to control the noise but also merely use consistent gradients for updation to learn the temporal-spatial dependency of generalization[46] to improve the price movement prediction.

Due to the two INTER modules are set to communicate the inter-task information, the sharing mechanism only influences their learnable parameters sets $W_{\text{yinter}}$ and $W_{\text{zinter}}$. Suppose we flatten the two parameters sets as two vectors (still denoted by $W_{\text{yinter}}$ and $W_{\text{zinter}}$ for simplicity), denote $j^{\text{th}}$ elements of the parameter vectors $W_{\text{yinter}}$ and $W_{\text{zinter}}$ after the $(i+1)^{\text{th}}$ updation by $p_{yj}^{i+1}$ and $p_{zj}^{i+1}$ respectively and their corresponding latest gradient elements before the $(i+1)^{\text{th}}$ updation by $g_{yj}^{i+1} \in \mathbb{R}$ and $g_{zj}^{i+1} \in \mathbb{R}$ respectively. Consequently, the new parameters of the two tasks after the $(i+1)^{\text{th}}$ updation $p_{yj}^{i+1}$ and $p_{zj}^{i+1}$ are calculated as follows:

$$p_{yj}^{i+1} = p_{yj}^{i} - \eta g_{yj}^{i+1} \mathbb{1}\{g_{yj}^{i+1} g_{zj}^{i+1} > 0\} \tag{11}$$

$$p_{zj}^{i+1} = p_{zj}^{i+1} - \eta g_{zj}^{i+1} \mathbb{1}\{g_{yj}^{i+1} g_{zj}^{i+1} > 0\} \tag{12}$$

where $\eta$ (e.g. 0.01) is the learning rate for the INTER module.

It is common that the rules with the generalization is learned from true values and the fake rules without generalization is learned from noise. For two highly related targets, it is reasonable to assume the true values of them to be consistent (the information learned from one target able to predict another target) and the noises of them to be inconsistent (the information learned from one target unable to predict another target)[46].

For the noisy prediction target (e.g. the additive noise, $y = \bar{y} + \epsilon$, $\bar{y}$ being true value and $\epsilon$ being the noise), it is likely that the model not only learns from the true value but also learns from the noise. Moreover, the gradients (or partial derivative) usually represent the information learned by the model and the updation of parameters is model learning. For the partial derivatives with the same sign value, we can find one direction to improve the prediction of two targets at the same time so that the updation of these parameters is more likely to learn the rules with generalization. For example, for the two negative partial derivatives, we can increase the corresponding parameters to improve the prediction of two targets and the consistency shows that the partial derivative is more likely learning the true value rather than the noise and is thus of generalization. However, for the partial derivatives with the different sign value, we cannot find one direction to improve the prediction of two targets at the same time so that the ignorance of updation of these parameters is more likely to filter the noise and improve generalization of models. For example, for the combination of one negative partial derivative and one positive partial derivative, we can increase the corresponding parameters to improve the prediction of one target but the prediction of another target will be damaged, which shows the inconsistency so that the two partial derivatives are more likely learning the noise rather than the true values and are thus of little generalization. If the noise of auxiliary target $r$ and the main target $y$ is high, the $y$ is closely related with $y$ and the input variable is able to explain the $y$ and $r$(e. g. $y=f(x)+\epsilon_y$ and $r=f(x)+\epsilon_r$, var($\epsilon_y$) and var($\epsilon_r$) are high), the two-target multitask method is better than directly taking the extra variable $r$ as input variable because $x$ is able to explain $r$, taking $r$ as the input variable is less likely to provide more new information to predict $y$ but will induce more noise to increase the difficulties of modeling. However, because the relation between $y$ and $r$ is close, the sharing part of the two-target method is able to identify and filter the noise learned from $y$ and $r$ and retain the information learned, which enables the model to learn the information with generalization rather than the noise without generalization and eventually improve the generalization of the model. Because the high-frequency price movement and rate of price change is closely related(see Table 1), the two-target multitask method taking the high-frequency rate of price change as an auxiliary target is better than taking it as direct input.

In addition, the multiplication of the indicator function is similar to the dropout method[59], both choosing some parameters absent from the parameters updation, but different from the dropout method, which chooses the parameters in a random way, the proposed method chooses the parameters based on the relation between the main target (price movement) and the auxiliary target(rate of price change) so that the sharing part of the proposed method can be seen as a special dropout method, which is valid to prevent overfitting[59] to improve the price movement prediction.

## 3 Experiments

This section evaluates the performance of the proposed method by the experiments on real-world datasets. In addition, LightGBM[60], CNN, LSTM and the direct input method(see Section 3.2.4) are used as baselines. For simplicity, we use the AuxIn to represent the direct input method and the AuxOut to represent the proposed auxiliary target method.

### 3.1 Description of datasets

In order to evaluate the performance of the proposed method for high-frequency trading, we conduct experiments on both future market and stock market. Specifically, we use the one-minute snapshot of the transaction data from 2014-3-21 to 2020-3-30 of three kind of Chinese futures (HC future, I future and J future), with each dataset containing the name of the future contract, the transaction date, intraday transaction time, open price, highest price, lowest price, close price, volume and position. Besides, we use the one-minute snapshot of the transaction data from 2019-1-1 to 2019-12-31 of three American stocks (AAPL and NLFX), with each dataset containing the transaction date, intraday transaction time, open price, highest price, lowest price, close price and volume.

We firstly transform the raw dataset to the sliding-window format and normalize (zero mean and one variance), then separate samples after specific date (e. g. 2020-1-1 for three futures and 2019-11-1 for two stocks) as the testing dataset, ten percent of the rest samples are used as validation dataset and the rest ninety percent used as the training dataset because the number of parameters is so huge that more traning data is needed.

Table 2 presents the number and ratio of three classes of the training, validation and testing dataset of different futures and stocks. Although the ratios of the up class and the down class of different datasets are relatively close, ratios of the unchanged class are slightly lower than the up class and the down class on HC, J and AAPL datasets but the slightly higher on the I, NFLX and AMZN datasets, which demonstrates that the overall distributions of training, validation and testing dataset are not extremely imbalanced.

**Table 2.** Distributions of price movement on training, validation and testing datasets of different futures.

| Datasets | | up | | unchanged | | down | |
|---|---|---|---|---|---|---|---|
| | | number | ratio (%) | number | ratio (%) | number | ratio (%) |
| HC | training | 93027 | 35.67 | 74662 | 28.63 | 93113 | 35.70 |
| | validation | 10345 | 35.70 | 8312 | 28.68 | 10320 | 35.61 |
| | testing | 3394 | 38.09 | 2110 | 23.68 | 3406 | 38.23 |
| I | training | 89210 | 29.71 | 122068 | 40.65 | 89029 | 29.65 |
| | validation | 10021 | 30.03 | 13501 | 40.46 | 9845 | 29.51 |
| | testing | 3050 | 34.46 | 2707 | 30.59 | 3093 | 34.95 |
| J | training | 107995 | 36.95 | 76517 | 26.18 | 107782 | 36.87 |
| | validation | 12099 | 37.25 | 8422 | 25.93 | 11956 | 36.81 |
| | testing | 3509 | 39.38 | 1798 | 20.18 | 3603 | 40.44 |
| NFLX | trading | 22374 | 31.88 | 25418 | 36.22 | 22382 | 31.90 |
| | validation | 2574 | 33.01 | 2768 | 35.50 | 2455 | 31.49 |
| | testing | 3760 | 24.51 | 7710 | 50.26 | 3871 | 25.23 |
| AMZN | trading | 19152 | 27.29 | 31495 | 44.88 | 19528 | 27.83 |
| | validation | 2228 | 28.58 | 3365 | 43.16 | 2204 | 28.27 |
| | testing | 2778 | 18.1 | 9823 | 63.99 | 2749 | 17.91 |
| AAPL | trading | 30352 | 43.25 | 9926 | 14.15 | 29895 | 42.60 |
| | validation | 3348 | 42.94 | 1131 | 14.51 | 3318 | 42.55 |
| | testing | 5673 | 36.98 | 4178 | 27.23 | 5400 | 35.20 |

### 3.2 Compared methods

#### 3.2.1 LightGBM

This model tries to find the structure of the input data to predict the target based on gradient boosting decision tree[60]. The input of LightGBM is the one-dimension reshaped vector of the sliding-window data and

LightGBM treats the input variables independently, which means that it ignores the spatial dependency and temporal dependency of the raw financial variables.

### 3.2.2 CNN

It tries to learn temporal-spatial dependency of temporal-spatial invariance based on the local intersection and shared weights. Moreover, with the fixed sizes of the two-dimension filters of CNN smaller than the size of sliding-window data, CNN can extract the static and local temporal patterns. In addition, each output unit of the CNN involves information of all financial variables, which means that the features extracted by CNN are globally spatial. On the whole, the patterns extracted by CNN is statically and locally temporal and globally spatial. The input data is the sliding-window data and the target is the sole target, price movement.

### 3.2.3 LSTM

It tries to learn the sequential structure or the context information of input by calculating the hidden states sequentially. Due to each output unit of the LSTM involves information of all financial variables and all the time points, the features extracted by LSTM are globally temporal and globally spatial. In addition, the temporal patterns are dynamic because of the three gates and the sequential calculation. On the whole, the patterns extracted by LSTM is dynamically, sequentially and globally temporal and globally spatial. The input and target is same as the CNN baseline.

### 3.2.4 AuxIn

It is the traditional method to use a variable to predict the price movement by taking the variable as a input variable. The structure of the AuxIn is the same as the structure of the main task part of the proposed method (AuxOut) but it takes the auxiliary target as an extra input variable besides the original input variables. Although the AuxIn and AuxOut methods directly use the extra information of the auxiliary target to predict the price movement, the concrete ways are different so that we can know whether the proposed multitask method is more suitable to extract the extra information of the magnitude by comparing with the AuxIn.

## 3.3 Evaluation metrics

Accuracy rate and accumulative revenue are utilized to evaluate the performances of different methods. Given a specific minute, after getting the probability prediction for the three classes, the class of the highest probability is used as the final prediction at that time.

As the price movement prediction is a three-class classification task and the overall distributions of training, validation and testing dataset are not extremely imbalanced (see Table 2), the accuracy rate is suitable to evaluate the performances of models.

The accumulative revenue is used to evaluate the capability of making money. Specifically, we design a strategy to map the prediction to the accumulative revenue, that is to buy at the beginning of the $t^{\text{th}}$ minute and sell at the end of the $t^{\text{th}}$ minute when the prediction is positive, to sell at the beginning and buy at the end when negative and to not trade when zero. Besides, the accumulative revenue is the money at the end of the test period by trading following the above strategy with no initial capital. Besides, the transaction fee (e. g. 1 yuan per unit for future and 0.1% for stocks) is considered for the calculation of accumulative revenue. Particularly, the transaction fee is set two-side, which means that we should pay transaction fee whether to buy or to sell.

## 3.4 Experimental settings

All of the methods except LightGBM are trained based on PyTorch and the batchsize is 20% of the number of training samples. We train the models for at most 100 epochs, and choose the best parameters when the loss of the main task on the validation dataset is the smallest. As for the two-layer CNN baseline, the learning rate is 0.003 for three futures and 0.001 for two stocks, the number of filters of first layer is 128 and the number of filters of second layer is 64. As for LSTM baseline, the learning rate is 0.005 for three futures and 0.001 for two stocks, the number of layers is one and the size of the hidden units is 100. For the INTER module of the AuxIn, the learning rate is 0.005 for three futures and 0.001 for two stocks; the number of filters is 32; the parameters (S,M,L) for HC and I are set to (3,12,22) and set to (1,10,20) for J, NFLX and AAPL. For the INTRA module of the AuxOut, the learning rate is 0.01 for three futures and 0.001 for two stocks and the size of hidden states is 96. For the INTER module of the AuxOut, the learning rate is 0.005 for three futures and 0.001 for two stocks; the number of filters is 16 for the NFLX and 32 for rest datasets; the parameter (S,M,L) for HC is set to (5,15,25) and set to (1,10,20) for rest datasets. For the INTRA module of the AuxOut, the learning rate is 0.01 for three futures and 0.001 for two stocks and the size of hidden states is 16 for NFLX and 96 for rest datasets.

As for the two-layer CNN baseline, we tune the number of filters and learning rate, with the number of filters of the first layer being {32, 64, 128} and that of the second being {8, 16, 64} respectively, and the learning rate being {0.001, 0.003, 0.005}. At last, the 128 filters of the first layer, 64 filters of the second layer and the 0.003 learning rate is the best. As for the LSTM baseline, we tune the number of layers and size of hidden states, with number of layers being {1, 2}, with size of hidden states being {50, 100, 192}. It turns out that one-layer and 100 hidden units are the best. As for AuxIn and the AuxOut, we tune the number of filters being {8, 16, 32, 64}, the parameter being (S,M,L) {(1,10,20), (3,12,22), (5,15, 25)} of the INTER module and the size of hidden states being {16, 32, 64} of the INTRA module.

### 3.5 Experimental results and analysis

Table 3 shows the performances of different models. Through comparable analysis, we can get the following results.

**Table 3.** Comparisons of different methods.

| Dataset | Methods | acc(%) | Accumulative revenue | |
|---|---|---|---|---|
| | | | Without fee | With fee |
| HC | LightGBM | 42.33 | 1500.00 | 1232.70 |
| | CNN | 43.14 | 1598.00 | 1330.70 |
| | LSTM | 43.09 | 1639.00 | 1371.70 |
| | AuxIn | 43.25 | 1572.00 | 1304.70 |
| | AuxOut | **43.69** | **1793.00** | **1525.70** |
| I | LightGBM | 40.89 | 527.00 | 261.50 |
| | CNN | 41.85 | 501.00 | 235.50 |
| | LSTM | 43.20 | 754.00 | 488.50 |
| | AuxIn | 42.75 | 731.00 | 465.50 |
| | AuxOut | **43.29** | **793.00** | **527.50** |
| J | LightGBM | 44.84 | 774.00 | 506.70 |
| | CNN | 45.19 | 845.50 | 578.20 |
| | LSTM | 45.17 | 856.50 | 589.20 |
| | AuxIn | 44.96 | 902.50 | 635.20 |
| | AuxOut | **45.39** | **913.00** | **645.70** |
| NFLX | LightGBM | 49.34 | −65.44 | −91.67 |
| | CNN | **53.50** | 59.41 | 18.21 |
| | LSTM | 51.44 | **159.90** | **102.21** |
| | AuxIn | 53.11 | −84.05 | −110.04 |
| | AuxOut | 53.19 | 99.86 | 53.23 |
| AMZN | LightGBM | 62.31 | −391.08 | −496.62 |
| | CNN | 62.55 | 76.18 | −31.80 |
| | LSTM | 61.61 | −81.87 | −184.29 |
| | AuxIn | 63.06 | −81.47 | −171.28 |
| | AuxOut | **63.21** | **527.74** | **398.29** |
| AAPL | LightGBM | 42.62 | 149.87 | 35.46 |
| | CNN | 42.58 | **842.77** | −42.12 |
| | LSTM | 36.85 | 109.93 | 5.40 |
| | AuxIn | 39.00 | 19.22 | −59.51 |
| | AuxOut | **43.16** | 297.90 | **142.75** |

[Note] The bolded numbers are the highest metric values of five different methods of one specific dataset.

The prediction accuracy and accumulative revenue (with or without) of CNN and LSTM are higher than LightGBM in most cases so that CNN and LSTM perform better than LightGBM for high-frequency price movement prediction. This is probably because high-frequency financial data is temporal-spatial dependent so that CNN modeling static accumulative temporal dependency and LSTM modeling the dynamic sequential temporal patterns outperform LightGBM failing to consider the temporal-spatial dependency. These results also imply that the design of the feature extractor of the AuxOut is suitable for high-frequency price movement prediction because it is the combination of CNN and LSTM modeling the temporal-spatial dependency of the high-frequency financial data.

The AuxOut outperforms LightGBM, CNN and LSTM on the accuracy and the accumulative revenue metrics in most of cases, which shows the two-target multitask method considering the related auxiliary target is better than the single task method failing to consider the related auxiliary target. Because the relation between the main target (high-frequency price movement) and the auxiliary target (high-frequency rate of the price change) helps the two-target multitask method to find a more certain distribution of the main target (high-frequency price movement) so that the two-target multitask method shows more generalization and gets better prediction performance.

The AuxOut performs better than the AuxIn concerning the accuracy and the accumulative revenue metrics in all cases, which indicates that taking the high-frequency rate of price change as the auxiliary target is better than taking it as the direct input variable. Moreover, Table 4 adds that the closer the relation between the high-frequency price movement and the rate of the price change, the better the two-target multitask method performs than taking the extra variable as the direct input method.

**Table 4.** Intensity of relation of the two targets and the performance difference of the two-target method and directly taking the extract variable as input variable.

| Datasets | Correlation | Mutual information ratio(%) | | Δacc(%) |
|---|---|---|---|---|
| | | Mean | Median | |
| HC | 0.78 | 95.72 | 97.63 | 0.44 |
| I | 0.88 | 95.43 | 97.50 | 0.54 |
| J | 0.76 | 95.87 | 97.69 | 0.43 |
| NFLX | 0.68 | 95.52 | 97.59 | 0.08 |
| AMZN | 0.66 | 95.31 | 97.42 | 0.15 |
| AAPL | 0.68 | 96.07 | 97.83 | 4.16 |

[Note] The Δacc(%) is the prediction accuracy on testing dataset of the AuxOut (the two-target multitask method) from minus the prediction accuracy of the AuxIn (taking the extra variable as input variable method).

**Table 5.** Changes of accurate score of different modules after removing the auxiliary target.

| Datasets | INTER | INTRA | INTER & INTRA |
|---|---|---|---|
| HC | **−0.76** | **−2.50** | **−1.80** |
| I | **−0.89** | **−1.08** | **−1.18** |
| J | **−1.18** | **−0.80** | **−1.00** |
| NFLX | **−0.95** | **−5.63** | **−1.22** |
| AMZN | 1.60 | **−1.15** | 0.35 |
| AAPL | **−0.40** | **−0.33** | **−0.07** |

[Note] The bolded number indicates that the accurate score of the features extracted by the module declines after removing the auxiliary target.

### 3.6 Ablation study

To explore the effects of the auxiliary target on the different temporal-spatial dependency learned by different parts of feature extractor in more detail, we conduct the ablation study by removing the link mechanism manually and comparing the performances of different modules with those of the original AuxOut. Specifically, we take the outputs of different modules as the input features to train LightGBM model and then evaluate their performances. In addition, the output of INTRA module is its last hidden states, the output of INTER module is the concatenation of outputs of three convolutional neural networks before the sum aggregation layer, and the output of the method is the concatenation of the above two outputs.

Table 5 shows the changes of the prediction accuracy of the features of different modules after removing the auxiliary target. The auxiliary target can improve the generalization of the overall temporal-spatial dependency because the performances of feature combination of INTER & INTRA are worse after removing the auxiliary target. Moreover, the auxiliary target can improve the generalization of the static accumulative temporal dependency because the performances of feature of the INTER module are worse after removing the auxiliary target. Besides, the auxiliary target can improve the generalization of the dynamic sequential temporal dependency because the performances of feature of the INTRA module are worse after removing the auxiliary target.

## 4 Conclusions

The two-target multitask method, with coarse-grained high-frequency price movement as the main target and fine-grained rate of price change as the auxiliary target, is proposed to utilize the close relation between its two targets to filter the high noise of targets and learn the rules with generalization from the true value to improve the high-frequency price movement prediction. On one hand, a feature extractor is designed to learn diverse temporal-spatial dependency for further processing of other parts of the method to predict the price movement, which consists of two modules based on recurrent neural network and convolutional neural network respectively to learn the dynamic sequential and the static accumulative temporal-spatial dependency of the high-frequency transaction data. On the other hand, a gradient balancing approach is adopted to use the close relation between two targets to filter the temporal-spatial dependency learned from the inconsistent noise and retain the dependency learned from the consistent true information to improve the high-frequency price movement prediction. Experimental results demonstrate that the method outperforms all the baselines in most cases, which shows that the method manages to incorporate the related information between high-frequency price movement and rate of price change to improve the prediction accuracy of the price movement. Moreover, through comparing the performance of different modules with and without the auxiliary target, we find that the auxiliary target not only improves the generalization of overall temporal-spatial dependency learned by the whole feature extractor but also improve temporal-spatial dependency learned by the different parts of the feature extractor.

At last, we give some future research suggestions: ① to model more high-frequency data(e.g. five-minute data); ② to search more valid auxiliary target; ③ to design better feature extractors (e.g. deeper neural networks).

## Acknowledgments

This work is supported by National Natural Science Foundation of China (No.71873128).

## Conflict of interest

The authors declare no conflict of interest.

## Author information

**MA Yulian** is a master candidate at School of Management, University of Science and Technology of China. Her research field is financial engineering and quantitative investment.

**CUI Wenquan** (corresponding author) is an associate professor at School of Management, University of Science and Technology of China (USTC). He received the PhD degree of statistics from USTC. His research interests focus on multivariant survival analysis and machine learning.

## References

[1] Fama E F. Random walks in stock market prices. Financial Analysts Journal, 1995, 51(1): 75−80.

[2] Thaler R H. Behavioral economics: Past, present, and future. American Economic Review, 2016, 106(7): 1577−1600.

[3] Slovic P, Finucane M, Peters E, et al. Rational actors or rational fools: Implications of the effects heuristic for behavioral economics. Journal of Socio-Economics, 2002, 31(4): 329-342.

[4] Ariel R A. A monthly effect in stock returns. Journal of Financial Economics, 1987, 18(1): 161-174.

[5] Barry C B, Brown S J. Differential information and the small firm effect. Journal of Financial Economics, 1984, 13(2): 283-294.

[6] Bustos O, Pomares-Quimbaya A. Stock market movement forecast: A systematic review. Expert Systems with Applications, 2020, 156: 113464.

[7] Ma Z, Bang G, Wang C, et al. Towards earnings call and stock price movement. https://arxiv.org/abs/2009.01317.

[8] Qiu M, Li C, Song Y. Application of the artificial neural network in predicting the direction of stock market index. In 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS 2016). IEEE, 2016: 219-223.

[9] Rustam Z, Nurrimah, Hidayat R. Indonesia composite index prediction using fuzzy support vector regression with fisher score feature selection. International Journal on Advanced Science, Engineering and Information Technology, 2019, 9(1): 121-128.

[10] Yao S, Luo L, Peng H. High-frequency stock trend forecast using LSTM model. In 2018 13th International Conference on Computer Science Education (ICCSE). IEEE, 2018: 1-4.

[11] Lim Y S, Gorse D. Deep probabilistic modelling of price movements for high-frequency trading. In 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020: 1-8.

[12] Feng F, Chen H, He X, et al. Enhancing stock movement prediction with adversarial training. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19). Macao, China: International Joint Conferences on Artificial Intelligence Organization, 2019: 5843-5849.

[13] Pagolu V S , Reddy K N , Panda G, et al. Sentiment analysis of Twitter data for predicting stock market movements. In 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES). IEEE, 2016: 1345-1350.

[14] Shi L, Teng Z, Wang L, et al. Deepclue: Visual interpretation of text-based deep stock prediction. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(6): 1094-1108.

[15] Zhang X, Zhang Y, Wang S, et al. Improving stock market prediction via heterogeneous information fusion. Knowledge-Based Systems, 2018, 143: 236-247.

[16] Chen M Y, Chen T H. Modeling public mood and emotion: Blog and news sentiment and socio-economic phenomena. Future Generation Computer Systems, 2019, 96: 692-699.

[17] Mohan S, Mullapudi S, Sammeta S, et al. Stock price prediction using news sentiment analysis. In 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService). IEEE, 2019: 205-208.

[18] Shah D, Isah H, Zulkernine F. Predicting the effects of news sentiments on the stock market. In 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018: 4705-4708.

[19] Tsantekidis A, Passalis N, Tefas A, et al. Using deep learning to detect price change indications in financial markets. In 2017 25th European Signal Processing Conference (EUSIPCO). IEEE, 2017: 2511-2515.

[20] Wang J, Sun T, Liu B, et al. Clvsa: A convolutional LSTM based variational sequence-to-sequence model with attention for predicting trends of financial markets. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19). Macao, China: International Joint Conferences on Artificial Intelligence Organization, 2019: 3705-3711.

[21] Gudelek M U, Boluk S A, Ozbayoglu A M. A deep learning based stock trading model with 2-D CNN trend detection. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2017: 1-8.

[22] Dang Q V. Reinforcement learning in stock trading. In Advanced Computational Methods for Knowledge Engineering. ICCSAMA 2019. Cham, Switzerland: Springer, 2020: 311-322.

[23] Mahdisoltani F, Memisevic R, Fleet D. Hierarchical video understanding. In Computer Vision-ECCV 2018 Workshops. ECCV 2018. Cham, Switzerland: Springer, 2019: 659-663.

[24] Campbell J Y, Hentschel L. No news is good news: An asymmetric model of changing volatility in stock returns. Journal of Financial Economics, 1992, 31(3): 281-318.

[25] Li Z, Tam V. A comparative study of a recurrent neural network and support vector machine for predicting price movements of stocks of different volatilites. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2017: 1-8.

[26] Basak S, Kar S, Saha S, et al. Predicting the direction of stock market prices using tree-based classifiers. North American Journal of Economics and Finance, 2019, 47: 552-567.

[27] Klein T. Trends and contagion in WTI and Brent crude oil spot and futures markets: The role of OPEC in the last decade. Energy Economics, 2018, 75: 636-646.

[28] Caruana R, De Sa V. Promoting poor features to supervisors: Some inputs work better as outputs. In Advances in Neural Information Processing Systems, Volume 9. Cambridge, MA: MIT Press, 1997.

[29] Ul Haq A , Zeb A, Lei Z, et al. Forecasting daily stock trend using multi-filter feature selection and deep learning. Expert Systems with Applications, 2021, 168: 114444.

[30] Dai C, Lu K, Xiu D. Knowing factors or factor loadings, or neither? Evaluating estimators of large covariance matrices with noisy and asynchronous data. Journal of Econometrics, 2019, 208(1): 43-79.

[31] Hahn J, Yoon H. Determinants of the cross-sectional stock returns in Korea: Evaluating recent empirical evidence. Pacific-Basin Finance Journal, 2016, 38: 88-106.

[32] Kohara K. Selective-learning-rate approach for stock market prediction by simple recurrent neural networks. In Knowledge-Based Intelligent Information and Engineering Systems. KES 2003. Berlin: Springer, 2003: 141-147.

[33] Zheng Q, Zhu F, Qin J, et al. Multiclass support matrix machine for single trial EEG classification. Neurocomputing, 2018, 275: 869-880.

[34] Antweiler W, Frank M Z. Is all that talk just noise? The information content of internet stock message boards. Journal of Finance, 2004, 59(3): 1259-1294.

[35] Nti I K, Adekoya A F, Weyori B A. Efficient stock-market prediction using ensemble support vector machine. Open Computer Science, 2020, 10(1): 153-163.

[36] Li C, Song D, Tao D. multitask recurrent neural networks and higher-orderMarkov random fields for stock price movement prediction. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2019: 1141-1151.

[37] Caruana R A. Multitask connectionist learning. In Proceedings of the 1993 Connectionist Models Summer School. New York: Psychology Press, 1993: 372-379.

[38] Ruder S. An overview of multitask learning in deep neural networks. http://export.arxiv.org/pdf/1706.05098.

[39] Mills T C, Markellos R N. The Econometric Modelling of Financial Time Series. Cambridge: Cambridge University Press, 2008.

[40] Sun J, Xiao K, Liu C, et al. Exploiting intra-day patterns for market shock prediction: A machine learning approach. Expert Systems with Applications, 2019, 127: 272-281.

[41] Hirchoua B, Ouhbi B, Frikh B. Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy. Expert Systems with Applications, 2021, 170: 114553.

[42] Fang J, Xia S, Lin J, et al. Alpha discovery neural network based on prior knowledge. https://arxiv.org/abs/1912.11761.

[43] Blumer A, Ehrenfeucht A, Haussler D, et al. Occam's Razor. Information Processing Letters, 1987, 24(6): 377-380.

[44] He K, Wang Z, Fu Y, et al. Adaptively weighted multitask deep network for person attribute classification. In MM 2017: Proceedings of the 2017 ACM Multimedia Conference. New York: Association for Computing Machinery, 2017: 1636-1644.

[45] Tang W, Wu Y. Does learning specific features for related parts help human pose estimation? In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2019: 1107-1116.

[46] Chen Z, Ngiam J, Huang Y, et al. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In Advances in Neural Information Processing Systems 33 (NeurIPS 2020). San Diego, CA: Neural Information Processing Systems, 2020.

[47] Yu Y, Si X, Hu C, et al. A review of recurrent neural networks: LSTM cells and network architectures. Neural Computation, 1019, 31(7): 1235-1270.

[48] Su H, Qi C R, Li Y, et al. Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. In Proceedings of the IEEE International Conference on Computer Vision. IEEE, 2015: 2686-2694.

[49] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[50] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735-1780.

[51] Levinshtein A, Sereshkeh A R, Derpanis K G. Datnet: Dense auxiliary tasks for object detection. In 2020 IEEE Winter Conference on Applications of Computer Vision (WACV 2020). IEEE, 2020: 1408-1416.

[52] Wang J, Wang Q, Zhang H, et al. Sparse multiview task-centralized ensemble learning for ASD diagnosis based on age- and sex-related functional connectivity patterns. IEEE Transactions on Cybernetics, 2019, 49(8): 3141-3154.

[53] Ditthapron A, Banluesombatkul N, Ketrat S, et al. Universal joint feature extraction for P300 EEG classification using multitask autoencoder. IEEE Access, 2019, 7: 68415-68428.

[54] Yoo B, Kwak Y, Kim Y, et al. Deep facial age estimation using conditional multitask learning with weak label expansion. IEEE Signal Processing Letters, 2018, 25(6): 808-812.

[55] Zhang Yu, Yang Qiang. A survey on multitask learning. https://arxiv.org/abs/1707.08114.

[56] Tang H, Liu J, Zhao M, et al. Progressive layered extraction (PLE): A novel multitask learning (MTL) model for personalized recommendations. In RecSys 2020: 14th ACM Conference on Recommender Systems. New York: Association for Computing Machinery, 2020: 269-278.

[57] Ding F, Luo C. An adaptive financial trading system using deep reinforcement learning with candlestick decomposing features. IEEE Access, 2020, 8: 63666-63678.

[58] Zhong S, Pu J, Jiang Y G, et al. Flexible multitask learning with latent task grouping. Neurocomputing, 2016, 189: 179-188.

[59] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 2014, 15: 1929-1958.

[60] Ke G, Meng Q, Finley T, et al. LightGBM: A highly efficient gradient boosting decision tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc., 2017: 3147-3155.

# 一种预测高频价格的端到端双目标多任务方法

马玉莲[1,2]，崔文泉[1,2*]

1. 中国科学技术大学国际金融研究院，安徽合肥 230601；

2. 中国科学技术大学管理学院，安徽合肥 230026

*通讯作者. E-mail：wqcui@ustc.edu.cn

**摘要：**高频价格变动预测是预测价格在短时间内（比如1 min内）的变化方向（上涨、不变或下跌）. 用历史的高频交易数据去预测价格变化是一个比较困难的任务，这是因为二者之间的关系是高噪声、非线性和复杂的. 为提高高频价格预测准确率，提出了一个端到端的双目标多任务方法. 该方法引进了一个辅助目标（高频价格变化率），它和主目标（高频价格变化方向）是高度相关的并且能够提高主目标的预测准确率. 此外，每一个任务都有一个基于循环神经网络和卷积神经网络的特征提取模块，它可以学习出历史交易数据和两个目标之间的高噪声、非线性和复杂的时空相依关系. 为了缓解多任务方法的潜在的负迁移问题，每个任务的任务间共享部分和任务特有部分被显式地分开. 而且，通过一种梯度平衡方法利用两个目标之间的高相关性过滤掉从不一致性中学到的噪声的同时保留从一致性中学到的相依规律，从而提高高频价格变化方向预测准确率. 在真实数据集上的实验结果表明：所提方法能够利用高度相关的辅助目标帮助主任务的特征提取模块去学习出更有泛化能力的时空相依规律，最终提高高频价格变化方向预测准确率. 此外，辅助目标（高频价格变化率）不仅能够提高特征提取模块的总体效果，而且也提高特征提取模块的不同部分的效果.

**关键词：**多任务学习；细粒度辅助目标；特征提取；共享方法；负迁移；高频价格动态预测

(Continued from p. 195)

## Author information

**LI Fenglin** (corresponding author) is a PhD candidate at Department of Mathematics, University of Science and Technology of China. His research area is singularity theory and algebraic geometry.

## References

[1] Dimca A, Papadima S. Hypersurface complements, Milnor fibers and higher homotopy groups of arrangments. Annals of Mathematics, 2003, 158: 473-507.

[2] Dimca A. Hyperplane Arrangements: An introduction. Berlin: Springer, 2017.

[3] Orlik P, Terao H. Arrangements of Hyperplanes. Berlin: Springer, 1992.

# 具有较小的最高阶 Betti 数的超平面配置补空间

李凤麟[*]

中国科学技术大学数学科学学院，安徽合肥 230026

*通讯作者. E-mail：fenglin125@126.com

**摘要：**使用删除限制方法对补空间最高阶 Betti 数较小的超平面配置进行了分类.

**关键词：**超平面配置；删除限制方法；Betti 数