Journal of Circuits, Systems, and Computers
Vol. 30, No. 7 (2021) 2150122 (18 pages)
© World Scientific Publishing Company
DOI: 10.1142/S021812662150122X

# Stock Turnover Prediction using Search Engine Data*

Zhijin Wang[†,§], Yaohui Huang[‡,¶], Bing Cai[†,‖], Rui Ma[†,**]
and Zongyue Wang[†,††]

[†]*Computer Engineering College, Jimei University,*
*Yinjiang Road 185, Xiamen 361021, P. R. China*

[‡]*Chengyi University College, Jimei University,*
*Jimei Road 199, Xiamen 361021, P. R. China*
[§]*zhijin@jmu.edu.cn*
[¶]*yhhuang5212@gmail.com*
[‖]*ccailearning@gmail.com*
[**]*ruima@jmu.edu.cn*
[††]*wangzongyue@jmu.edu.cn*

Received 17 July 2020
Accepted 15 September 2020
Published 5 December 2020

The stock turnover values are sensitive to external factors, and remain great challenges in its prediction. The consideration is that search engine data can reflect market environment, policies and attentions on stocks. Therefore, a dual sides autoregression (DSAR) method is proposed to benefit from both observed turnover values and exogenous data. The proposed DSAR consists of linear representation stage and combination stage. In linear representation stage, the short-term patterns of turnover values and query data are represented, respectively. In combination stage, the outputs from previous stages are combined. Intensive experiments on two groups of data collections show the effectiveness of our proposed method.

*Keywords*: Stock turnover; search engine data; prediction; time series representation and consolidation.

## 1. Introduction

Due to the potential business value and research challenges on stock turnover forecasting, the stock turnover prediction is extensively focused by securities issuers and academic researchers. Stock turnover values reflect the activities of the stock market through the huge amount of statistical records.

There are two major considerations in predicting the stock turnover value in the future. First, the numeric value of a stock turnover value is usually very large and

*This paper was recommended by Regional Editor Tongquan Wei.
[††]Corresponding author.

varies greatly, which makes the prediction become harder. Second, the stock turnover values are easily affected by external environments.

To generate more accurate predictions by leveraging external information, lots of time series prediction methods using external information were investigated such as the NARX-RNN[1] which directly concatenates exogenous information and historical target observations as inputs of it. Deep neural networks are developed to fuse those information as well.[2,3] The commonly used exogenous data are inventories,[4] GDP[5] and unemployment rate.[6]

Stock-related information can be roughly divided into two categories: quantitative data and qualitative description.[7,8] Quantitative data includes but not limited to stock turnover value, stock price and standard income. The quantitative analysis of investors makes decisions based on published quantitative data. Qualitative data includes social appraisal, products and strategies of a company. The qualitative analysis of investors makes decisions based on company financial reports, short-term related news and national policies. All of the qualitative data can be directly found in search engines.

Both quantitative information and qualitative information are important to develop a successful investment strategy, and search engines play a significant role in accessing those information. Hence, the search index of a given keyword is taken into account, which can quantize the popularity of a company or a stock.

We propose a dual sides autoregression (DSAR) method for forecasting the stock turnover time series, via fusing the search engine data and historical stock turnover values. The proposed DSAR consists of a representation stage and a combination stage. In the representation stage, the short-term patterns of turnover values and query data are represented, respectively. In the combination stage, the outputs from the dual stages are combined. More detail, the DSAR leverages dual linear components to discover local dependency patterns among heterogeneous inputs and targets. An effective dual sides processing structure is considered to capture effects from past stock turnover values and Baidu search index, respectively. This structure refines the representation of inputs, and uses a dense layer to combine the outputs from dual represented sides of two kinds of inputs.

The main contributions of this work can be summarized as follows:

(i) We propose a stock turnover value prediction method by integrating dual sources of heterogeneous data. Compared with traditional methods, this method considers the joint impacts of social attention on investment decisions.
(ii) To alleviate the problem of data uncertainty and utilize the historical patterns, we explore the correlations among historical stock turnover values with segmenting. Instead of a simple linear combination of the exogenous features, we consider the delay effects among features to capture their correlations.
(iii) DSAR method is evaluated on two real stock data collections in the China A-share stock market, and the results show that our method can achieve

state-of-the-art than other traditional machine learning methods and classical deep learning methods. Compared to the state-of-art methods, DSAR not only demonstrates superiority in performance but also requires fewer parameters to tune.

The remainder of this paper is summarized as follows. Section 2 introduces related work. Section 3 illustrates our proposed method. Section 4 gives experimental configurations for fair comparison. Section 5 analyzes evaluated results. Finally, a conclusion is drawn in Sec. 6.

## 2. Related Work

This section introduces related stock prediction techniques. According to the type of input variable(s) of a method, these techniques are categorized into *univariate methods*[9–40] and *multivariate methods*.[41–48]

### 2.1. *Univariate methods*

The univariate methods predict future stock turnover values on the basis of past values. These methods can be divided into statistical methods,[9–14] learning methods,[15–34] and decomposition methods.[35–40]

The *statistical methods* include but not limited to GARCH,[9] EGARCH,[10] autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA).[11] ARIMA and its different variations based on the famous Box–Jenkins principle,[12] hence these methods are also popularly known as the Box–Jenkins models. However, the stock time series are inevitably corrupted by objective factors in the real environment, the historical observed values exhibit well-known nonstationary and uncertain features. Methods in this sub-category do not work well with nonstationary series.[13,14]

The *learning methods* aim to learn a linear or nonlinear mapping from known observations to the coming values. These methods are generally divided into traditional learning methods and temporal concerned learning approaches. The traditional learning methods consider an observed value as an input dimension, and a coming value as an output dimension. These methods include but not limited to multiple linear regression (MLR),[15] support vector machine regression (SVR),[16,17] gradient boosting decision tree regression (GBR),[18,19] random forest regression (RFR)[20] and extreme gradient boosting (XGB),[21] which implement machine learning algorithms under the gradient boosting framework. The temporal concerned methods consider the inherent temporal dynamics of historical values when predicting upcoming values. These methods include but not limited to Deep Multilayer Perceptron (DMLP), Recurrent Neural Network (RNN),[22] Long- and Short-Term Memory (LSTM),[23] Gated Recursive Network (GRU),[24] Convolutional Neural Networks (CNN),[25] Restricted Boltzmann Machines (RBM),[26] Deep Belief Networks

(DBN),[27] Autoencoder (AE)[28] and Deep Reinforcement Learning (DRL).[29] In addition, evolutionary computations (EC),[30] genetic algorithms (GA),[31] agent-based methods and multi-objective Evolutionary Algorithms (MOEA)[32–34] are extensively surveyed on various financial applications including financial time series prediction. However, these methods are easily remembering all the trained samples on the small-scale asset data, and have poor abilities of generalization.

For *decomposition methods*, a time series can be decomposed into several sub-series of different frequencies, e.g., empirical mode decomposition (EMD),[35–37] CEEMDAN[38] and RobustSTL.[39,40] EMD is a Fourier transform-based signal decomposition method, which can adaptively process any nonlinear and nonstationary signals. It should be noted that not all sequences can be well decomposed when a time series data is discrete and uncertain. Meanwhile, the stock time series data is easily affected by social environment and political decision.

### 2.2. *Multivariate methods*

To alleviate the problem of data uncertainty, kinds of exogenous data are collected and fused into the statistical methods[41] and learning methods.[43–48]

The *statistical methods* linearly combine past values of the target variable and exogenous variables to predict upcoming target values. Their key differences among these methods are regressions on the target variables, functions on exogenous data, and the composition of exogenous data. The most common method is ARIMAX,[41] which is also a different variation of ARIMA.

The *learning methods* in this category are divided into three sub-categories according to their model structures. (a) The traditional machine learning methods. Each exogenous value is an input dimension; (b) The temporal concerned methods. The temporal dynamics of input data are captured by using RNN structures, and a nonlinear mapping from inputs to the output is learned from training samples. For example, NARX-RNN enhances vanilla RNN using additional exogenous features, and differently treats exogenous inputs and target inputs.[1] DWNN is the combination of RNN and CNN.[42] SFM implements state-frequency in RNN to explore multiple frequency sequential patterns.[3] The encoder–decoder considers time series forecasting as sequence-to-sequence prediction.[43] (c) The attention mechanism methods. Typically, DA-RNN represents the input attention and temporal attention to extract significant exogenous inputs and past stock price inputs over time.[44] HRHN uses hierarchical attention mechanism to capture import details from inputs,[45] TPA-LSTM,[46] MTNet,[47] and LSTNet[48] are variant methods.

We focus on predictions based on turnover index data and their relevant query data. The assumption is that search index would reflect the social attention of a company, and indirectly influences the stock turnover. Hence, predictions based on past turnover values and query data has potentials in improving prediction performance. However, existing methods are designed for continuous variables, but the stock data are connected to discrete variables and has high real-time requirements.[49–51] These methods usually have poor performance in stock turnover prediction.

## 3. Our Approach

This section gives problem definition, and illustrates the proposed DSAR.

### 3.1. *Problem definition*

The problem of stock turnover prediction can be addressed as the problem of time series forecasting. Moreover, the problem of turnover prediction using exogenous data can be viewed as learning a nonlinear mapping from past turnover values and exogenous observations to the upcoming stock turnover. This is formulated as

$$\hat{y}_{T+1} = F(y_1, y_2, \ldots, y_T, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)\,, \tag{1}$$

where $\hat{y}_{T+1}$ is the upcoming values, $y_t \in \mathbb{R}$ denotes the observation measured at time $t$, $\mathbf{x}_t \in \mathbb{R}^n$ denotes exogenous factors at time $t$, and $F(\cdot)$ is a nonlinear mapping which should be learned.

Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ denote the observed query data in a time-span of size $T$, and $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ be the past logarithmic stock turnover values.

### 3.2. *Dual sides autoregression (DSAR)*

The diagram of the proposed DSAR is shown in Fig. 1. This diagram consists of three stages: data pre-processing and post-processing, search engine data representation and turnover data representation.

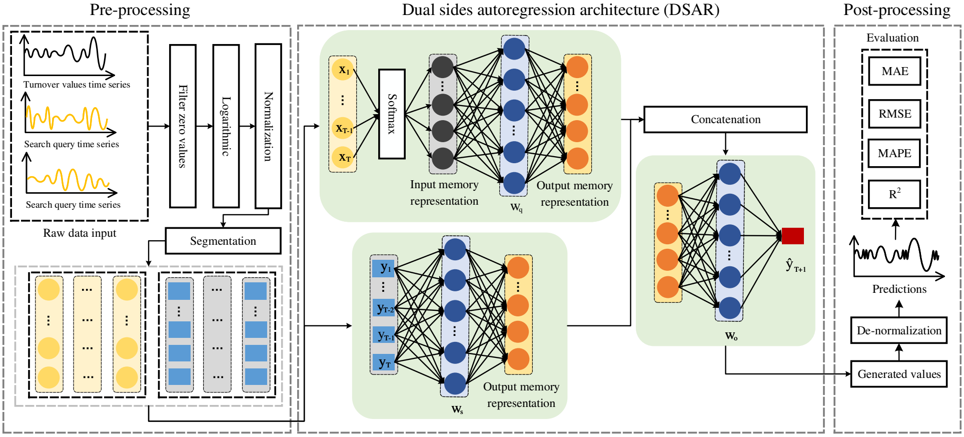


Fig. 1. The diagram of our proposed DSAR.

In the data pre-processing stage, as shown in the left part of Fig. 1, the inputs are normalized to do further segmentation. The inputted data consist of logarithmic search query data and logarithmic turnover data. In the right part of Fig. 1, the query data and turnover data representation stage, a softmax layer is exploited to reduce the magnitude disparity between strong and weak correlation search query data, and two linear layers are used to exploit series feature details and enhance the feature representation of input data, respectively. Finally, the dual represented results are linearly consolidated to generate predictions.

### 3.2.1. *Data pre-processing and post-processing*

*Normalization and de-normalization.* The normalization operation scales the data from the original range to another range. Due to the significant magnitude difference between search query data and turnover data, normalization is essentially required.

Both Min–Max normalization and standard (a.k.a, z-score) normalization are commonly applied to normalize time series data. Min–Max normalization scales data in the $[0, 1]$ interval by using the bounds of the minimum and maximum values. Standard normalization scales a dataset to a Gaussian distribution, so that the mean of observed values is 0 and the standard deviation is 1. The mean and standard deviation estimates of a dataset can be more robust to new data than the minimum and maximum values.

Min–Max normalization is considered to scale inputs and the target in the study, since the linear transformation of the original data can maintain the value differences. The min–max normalization of inputs is formulated and recovered as

$$\boldsymbol{d}' = \frac{\boldsymbol{d} - \min(\boldsymbol{d})}{\max(\boldsymbol{d}) - \min(\boldsymbol{d})}\,, \tag{2}$$

$$\boldsymbol{d} = \boldsymbol{d}' \cdot (\max(\boldsymbol{d}) - \min(\boldsymbol{d})) + \min(\boldsymbol{d})\,, \tag{3}$$

where $\boldsymbol{d} \in \mathbb{R}^M$ denotes a feature of observed samples, $M$ is the number of observed samples, $\boldsymbol{d}'$ is the normalized data, $\min(\cdot)$ is the minimal value of $\boldsymbol{d}$, and $\max(\cdot)$ is the maximal value of $\boldsymbol{d}$. The de-normalization formula is applied to revise the final predictions, which are generated by models.

*Segmentation.* The segmentation is the transformation of a time series to supervised data. To represent the short and medium-term temporal patterns from inputted time series, the time series data is transformed into data pairs of inputs and output and then the supervised methods are applied on these data pairs.

### 3.2.2. *Search query data representation*

*Input memory representation.* The correlation degree between various external data and the target data dynamically changes. The weakly correlated features may have poor effects on prediction, and the fluctuation of external data may affect the model fitting.

To ensure that the potential details from search queries are extracted from the DSAR. A softmax layer is exploited to reduce the magnitude disparity between strong and weak correlation data, and it is formulated as

$$\boldsymbol{p} = \mathrm{softmax}([\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]), \tag{4}$$

where the vector $\boldsymbol{p} \in \mathbb{R}^{n \times T}$ is viewed as the attention weight distributions on the memory inputs. The element in $\boldsymbol{p}$ can be represented as follows:

$$p_{m,t} = \frac{\exp(\mathbf{x}_{m,t})}{\sum_{j=1}^{N} \exp(\mathbf{x}_{j,t})}, \tag{5}$$

where $\exp(\cdot)$ represents the exponential function, and $\mathbf{x}_{j,t}$ represents the element in $\mathbf{x}$. The input memory $\boldsymbol{a}_t$ is the product of the input vector and the weight distribution, and is formulated as

$$\boldsymbol{a}_t = \boldsymbol{p} * [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T], \tag{6}$$

where the input vector $\mathbf{x}$ modified by the weighted distribution can better express the degree of correlation between search query data and turnover data.

*Output memory representation.* In order to represent the temporal features from the search queries after generating input memory representation, DSAR employs a linear layer to receive the internal association of last outputs.

Linear weighting highlights special periodic events and enhances the extraction of input information. The linear weighting of the input features can be formulated as

$$e_q = \sum \boldsymbol{w}_q * \boldsymbol{a}_t + b_q, \tag{7}$$

where $e_q \in \mathbb{R}$ is the weighted search query features matrix, $\boldsymbol{w}_q \in \mathbb{R}^{n \times T}$ is the weight corresponding to the input dimension, and $b_q$ is a bias term.

### 3.2.3. *Turnover data representation*

*Output memory representation.* The target time series (i.e., stock turnover time series) has a strong correlation among each time stamps. Different to the representation on exogenous data, the softmax layer is not used for transformation. In order to analyze the variation in stock data, a linear weighting component is retained as well as exogenous data representation. The linear weighting is formulated as

$$e_y = \boldsymbol{w}_y * [y_1, y_2, \ldots, y_T] + b_y, \tag{8}$$

where $e_y \in \mathbb{R}^1$ is the weighted search query features matrix, $\boldsymbol{w}_y \in \mathbb{R}^n$ is the weight corresponding to the input dimension, and $b_y$ is a bias term.

The proposed method combines search query and turnover values through a fully connected layer to correlate the outputs of dual sides. The combination of dual representations is formulated as

$$\hat{y}_{T+1} = \sigma(\boldsymbol{w}_o[e_y; e_q] + b_o), \tag{9}$$

where $[e_y; e_q] \in \mathbb{R}^2$ is the concatenated vector of dual sides outputs, $\boldsymbol{w}_o \in \mathbb{R}^2$ is the weight of outputs from dual represented sources, $\hat{y}_{T+1}$ is the predicted value of the logarithmic turnover values in the next weekday, and $b_o$ is a bias term, $\sigma$ represents the sigmoid function.

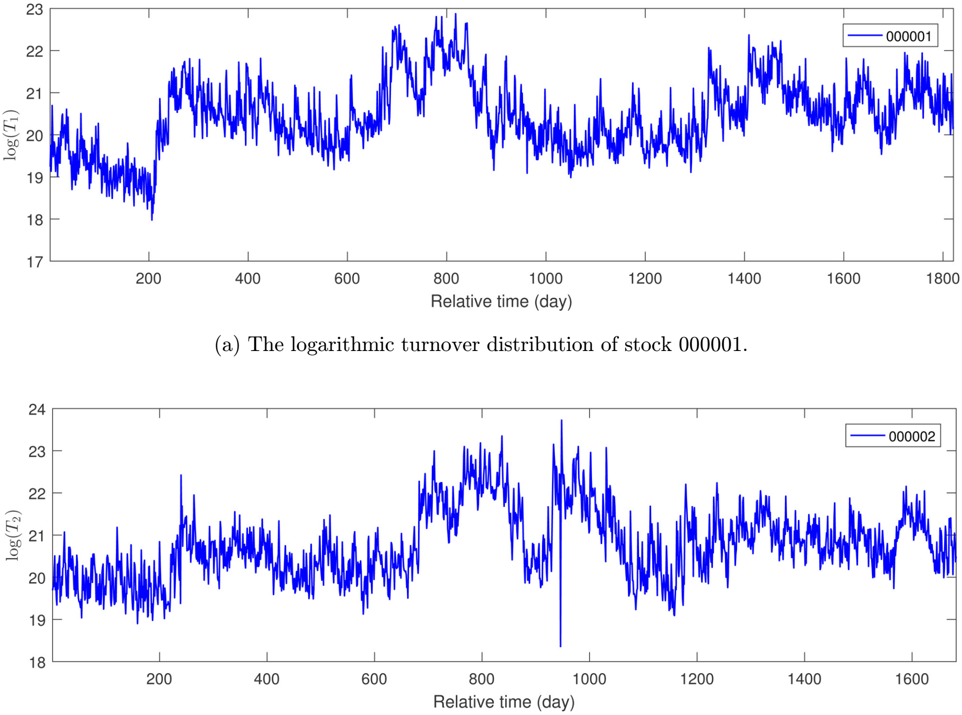
## 4. Experimental Setup and Benchmarks

This section gives study materials, evaluation metrics and benchmark methods.

### 4.1. *Turnover data and search engine data*

The China A-sharestock market is the main domestic stock sharing market in China, where stocks are subscribed and traded in China Yuan (CNY). We collected the historical stock data.

As plotted in Fig. 2, the stock dataset is collected during the period from 4 January 2011 to 18 July 2019, which is available public. It contains two companies and is organized in days. The turnover values of stock 000001 (Ping An Bank Co.,



(a) The logarithmic turnover distribution of stock 000001.

(b) The logarithmic turnover distribution of stock 000002.

Fig. 2. Turnover distributions of stock 000001 and 000002.

Ltd.) consist of 2057 observations, and stock 000002 (China Vanke Co., Ltd.) consists of 1924 observations.

Logarithmic BSI values and turnover values are used as the input data in the experiment, which reduces the difficulty of model training for large-magnitude numeric values. The dynamic deduction on those logarithmic values can more intuitively show the changes of turnover values in days, which helps to visualize the future development of trading volume.

The basic statistical characters of observations and their logarithmic values are listed in Table 1.

### 4.2. *Evaluation metrics*

The evaluation metrics are combined with mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE) and correlation coefficient ($R^2$).[52] These criteria can be expressed in the following mathematical expressions:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}(|y^{i}_{T+1} - \hat{y}^{i}_{T+1}|)\,, \tag{10}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y^{i}_{T+1} - \hat{y}^{i}_{T+1})^2}\,, \tag{11}$$

Table 1. The basic statistical features of inputs variables (BSI) and target variables (turnover values).

| Symbols | Min | Max | Medium | Mean | STD |
|---|---|---|---|---|---|
| $B_m^1$ | 352 | 27788 | 7780 | 7278.74526 | 4360.608619 |
| $B_d^1$ | 1511 | 37210 | 15905 | 16019.62227 | 6767.567975 |
| $B_t^1$ | 6555 | 61526 | 22266 | 23298.36753 | 8751.141306 |
| $T_1$ | 63,994,755 | 8,596,942,094 | 638,179,743 | 1,005,686,565 | 1,033,385,540 |
| $\log(B_m^1)$ | 5.863631 | 10.232360 | 8.959312 | 8.56347 | 0.988049 |
| $\log(B_d^1)$ | 7.320527 | 10.524333 | 9.674389 | 9.573979 | 0.490725 |
| $\log(B_t^1)$ | 8.787983 | 11.027215 | 10.010816 | 9.983075 | 0.389340 |
| $\log(T_1)$ | 17.974312 | 22.874672 | 20.274131 | 20.35216 | 0.848508 |
| $B_m^2$ | 455 | 92696 | 2238 | 3058.075884 | 4903.939952 |
| $B_d^2$ | 623 | 72302 | 2448 | 2999.287942 | 3403.818356 |
| $B_t^2$ | 1589 | 164998 | 4467 | 6057.363825 | 8212.152782 |
| $T_2$ | 93,815,937 | 20,106,488,583 | 857,364,772 | 1,387,510,703 | 1,608,417,397 |
| $\log(B_m^2)$ | 6.120297 | 11.437081 | 7.713114 | 7.675108 | 0.725525 |
| $\log(B_d^2)$ | 6.434547 | 11.188607 | 7.803027 | 7.861206 | 0.424866 |
| $\log(B_t^2)$ | 7.370860 | 12.013689 | 8.404472 | 8.506177 | 0.498378 |
| $\log(T_2)$ | 18.356845 | 23.724308 | 20.569374 | 20.640632 | 0.863284 |

*Note*: "STD" denotes the standard variation.

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{(|y_{T+1}^{i} - \hat{y}_{T+1}^{i}|)}{|y_{T+1}^{i}|} \times 100\%\,, \tag{12}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_{T+1}^{i} - \hat{y}_{T+1}^{i})^2}{\sum_{i=1}^{N}{y_{T+1}^{i}}^2}\,. \tag{13}$$

In the above equations, $y_{T+1}^{i}$ is the $i$th actual value in the test period, $\hat{y}_{T+1}^{i}$ is the $i$th predicted value, $N$ is the length of the test period. The performance with the smallest MAE, RMSE and MAPE and the largest $R^2$ are considered to be the best model.

### 4.3. *Benchmark methods*

This section gives several benchmark methods.

#### 4.3.1. *Multiple linear regression (MLR)*

MLR is widely used for modeling the linear relationship between the input variables and the target variable. The advantage is that the parameters do not need to be tuned.

#### 4.3.2. *Gradient boosting regression (GBR)*

GBR produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. A stable prediction model can be built by setting the number of boosting stages, learning rate, number of samples for each split, the minimum number of samples required to be at a leaf node and the maximum depth of the individual regression estimators.

#### 4.3.3. *Random forest regression (RFR)*

RFR is an ensemble learning method for regression via learning decision trees. The number of trees and a leaf node required the minimum number of samples can be set in order to adjust the model structure to the optimal state.

#### 4.3.4. *Extreme gradient boosting regression (XGboost)*

XGboost has gained much popularity and attention recently, which blends regularization term into loss function, and it is an improvement on the gradient boosting. The model is selected by the number of trees, learning rate, and maximum tree depth.

#### 4.3.5. *Recurrent neural network*

NARX-RNN exhibits temporal dynamic states of the sequence by their internal state. The hidden neurons are used to control the complexity of a RNN.

### 4.3.6. *Long- and short-term memory*

NARX-LSTM is a kind of recurrent neural network, which is composed of a cell, an input gate, an output gate and a forget gate. The number of hidden neurons is tuned to optimized the model.

### 4.3.7. *Gate recurrent unit (GRU)*

GRU is a variant of LSTM, which uses an update gate to replace the hidden and cell gates of LSTM. The GRU method adjusts hidden neurons to control the scale of neural network.

## 5. Results and Analyses

In this section, the experimental results are displayed to reveal the performance of algorithms by benefiting from the search engine data. Tables 2 and 3 summarize the evaluated results of all comparable methods over two kinds of inputs in terms of four metrics.

### 5.1. ***Main results***

The optimal MAE, RMSE, MAPE and $R^2$ values of comparable methods are found at $T = 10$. Hence, the performances of those algorithms are compared by fixing $T = 10$. Several important observations are made about these results:

(i) The proposed DSAR has the best performance over all the inputs in the four metrics.
(ii) By benefiting from search query data, all of comparable methods increase predict accuracy.

Table 2. Comparisons of different methods on stock 000001 and BSI data in terms of MAE, RMSE, MAPE, and $R^2$. $B^1 = \{B_t^1, B_m^1, B_d^1\}$.

| | $\log(T_1)$ | | | | $\log(T_1) + \log(B^1)$ | | | |
|---|---|---|---|---|---|---|---|---|
| Model | MAE | RMSE | MAPE | $R^2$ | MAE | RMSE | MAPE | $R^2$ |
| MLR | 0.292908 | 0.375703 | 0.013997 | 0.540597 | 0.293749 | 0.374669 | 0.014045 | 0.543121 |
| GBR | 0.300719 | 0.387361 | 0.014344 | 0.511644 | 0.301725 | 0.385408 | 0.014397 | 0.516557 |
| RFR | 0.308839 | 0.394645 | 0.014753 | 0.493105 | 0.309529 | 0.390883 | 0.014791 | 0.502724 |
| XGB | 0.302460 | 0.385955 | 0.014444 | 0.515182 | 0.301873 | 0.380885 | 0.014431 | 0.527837 |
| RNN-32 | 0.295733 | 0.381251 | 0.014113 | 0.526929 | 0.300457 | 0.377852 | 0.014394 | 0.535327 |
| RNN-64 | 0.295234 | 0.378088 | 0.014109 | 0.534746 | 0.294004 | 0.374793 | 0.014064 | 0.542819 |
| LSTM-32 | 0.296370 | 0.380188 | 0.014144 | 0.529562 | 0.291762 | 0.376522 | 0.013926 | 0.538593 |
| LSTM-64 | 0.296231 | 0.381628 | 0.014141 | 0.525992 | 0.293324 | 0.376842 | 0.014009 | 0.537806 |
| GRU-32 | 0.296861 | 0.379276 | 0.014174 | 0.531818 | 0.293579 | 0.376046 | 0.014025 | 0.539759 |
| GRU-64 | 0.295354 | 0.378535 | 0.014102 | 0.533644 | 0.294108 | 0.375321 | 0.014056 | 0.541530 |
| DSAR | | | | | **0.291038** | **0.371440** | **0.013909** | **0.550963** |

Table 3. Comparisons of different methods on stock 000002 and BSI data in terms of MAE, RMSE, MAPE, and $R^2$. $B^2 = \{B_t^2, B_m^2, B_d^2\}$.

| | $\log(T_2)$ | | | | $\log(T_2) + \log(B^2)$ | | | |
|---|---|---|---|---|---|---|---|---|
| Model | MAE | RMSE | MAPE | $R^2$ | MAE | RMSE | MAPE | $R^2$ |
| MLR | 0.264770 | 0.339071 | 0.012641 | 0.442676 | 0.266216 | 0.339758 | 0.012729 | 0.440416 |
| GBR | 0.276168 | 0.352292 | 0.013174 | 0.398366 | 0.277167 | 0.350270 | 0.013231 | 0.405253 |
| RFR | 0.274435 | 0.353815 | 0.013086 | 0.393154 | 0.270414 | 0.347660 | 0.012901 | 0.414083 |
| XGB | 0.274543 | 0.351755 | 0.013096 | 0.400200 | 0.275747 | 0.349166 | 0.013153 | 0.408997 |
| RNN-32 | 0.264676 | 0.338500 | 0.012639 | 0.444550 | 0.274475 | 0.347308 | 0.013132 | 0.415269 |
| RNN-64 | 0.264885 | 0.338863 | 0.012649 | 0.443359 | 0.268960 | 0.343304 | 0.012860 | 0.428674 |
| LSTM-32 | 0.267128 | 0.341731 | 0.012756 | 0.433898 | 0.263466 | 0.338300 | 0.012584 | 0.445206 |
| LSTM-64 | 0.267766 | 0.340929 | 0.012796 | 0.436551 | 0.266234 | 0.340774 | 0.012718 | 0.437063 |
| GRU-32 | 0.265628 | 0.339615 | 0.012682 | 0.440885 | 0.264181 | 0.337237 | 0.012627 | 0.448687 |
| GRU-64 | 0.264216 | 0.339253 | 0.012603 | 0.442077 | 0.265055 | 0.337619 | 0.012673 | 0.447440 |
| DSAR | | | | | **0.263396** | **0.336064** | **0.012581** | **0.452517** |

(iii) The performance of traditional linear type methods and recurrent neural network type methods are better than tree-based methods.

The results of comparable methods on stock 000001 are listed in Table 2. (1) MLR performs best when compared with other methods, which are solely based on logarithmic turnover index values. A possible reason is that the linear models captures the relevant temporal details, such as slumps, peaks and other temporal patterns. (2) The RNN, LSTM and GRU methods perform well. But when compared with MLR, they poorly perform on small scale data collections. The number of hidden neurons also has a impact on RNN models. RNN with more hidden neurons works better, but the results of LSTM is opposite to RNN. A possible reason is that RNN has a simple structure, with the increment of neurons, the structure fit the more mild fluctuation of stock 000001 and are uneasily to overfitting. Due to complex structure are easy to capture many details, which leads to local overfitting and degrades the prediction accuracy. (3) Reminder that tree-based methods, such as GBR, RFR and XGB have an unstable evaluation result, which suggests that most of the tree-based methods cannot relieve the random perturbations of logarithmic turnover time series, and unable to explore a stable pattern to map fluctuation of time series.

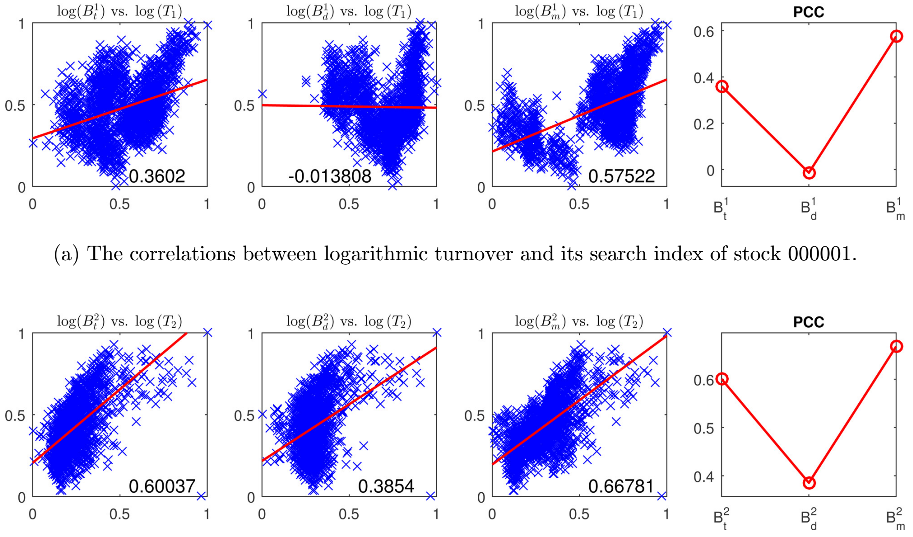
Intuitively, the performance differences between comparable methods are slight, i.e., in the magnitude of 0.001. A significant reason is that logarithmic turnover index values have a small variance, which limits the increment of prediction accuracy. Therefore, the accuracy of the prediction results is relatively low.

According to the prediction results based on the historical turnover index values and search query index in Table 2. (1) DSAR method shows the optimal evaluation results in terms of four metrics. (2) MLR shows preferable RMSE and MAE, LSTM performs better MAE and MAPE among the benchmark methods because RMSE is used to describe the dispersion of the loss between the predicted values and target values, while MAE and MAPE are applied to evaluate the absolute error between the predicted value and the target value. Therefore, the MLR method shows more balanced error and the LSTM method shows smaller absolute error in predicting the target value with larger value. (3) The prediction accuracy of all of the methods improve after fusing search query data, of which RNN, LSTM and GRU methods are the most obvious. The performance directly reflects that the fusion of search query data has a significant role in logarithmic turnover index prediction. The search query data records the historical changes of social concentration, which have the ability to capture the trend of specific stocks turnover and indirectly reduces the influence of turnover series random distributions for prediction methods.

The results of comparable methods on stock 000002 are listed in Table 3. (1) the overall prediction accuracy has a significant decline compared to stock 000001. There are two possible reasons. First, the data volume of stock 000002 is smaller than the other one. Second, stock 000002 has more drastic fluctuations than stock 000001, which leads to more difficulties in model fitting and reduced prediction accuracy. (2) According to the prediction results solely based on the historical turnover index values in Table 3, compared with the result of stock 0000001, RNN gets the optimal result, and the linear type methods show poorly performance based on the historical turnover index values. It suggests that the linear type model has defects for unstable series, the drastic random oscillations in series can easily cause the method to fail to converge to the best performance. Then the deep learning model has stronger anti-interference ability, which can reduce the interference to the prediction model. (3) The results of the tree-based methods perform poorly as before.

According to the prediction results based on the historical logarithmic turnover index values and search query data in Table 3, (1) DSAR maintains the optimal performance, which is significantly better than the other comparison methods. It is directly shown that DSAR has good stability and generalization performance. (2) Except the MLR and RNN, the prediction performance of all comparable methods has increased after fusing the search query data. A possible reason is that traditional linear type models have defects in processing complex time series data, and RNN has inherent shortages of gradient explosion and gradient disappearance, which magnifies in small scale data collections and complex series learning.

The DSAR method treats search query data and logarithmic turnover index data separately. First, it reduces the impact of random disturbance of search query index series on the original logarithmic turnover index values prediction, Second, it captures the details of search query data features well, and effectively integrates a variety of heterogeneous data for the upcoming prediction. Therefore, the prediction accuracy for two different stock data has been significantly improved.



(a) The correlations between logarithmic turnover and its search index of stock 000001.

(b) The correlations between logarithmic turnover and its search index of stock 000002.

Fig. 3. The visualization of correlations between the target variable and exogenous variables.

### 5.2. *Correlation analysis*

As shown in Fig. 3, the target variable is significantly correlated with logarithmic search query index variables, except the PC search query data of stock 000001. Their Pearson Correlation Coefficient (PCC) values are measured and plotted respectively, and compared in the last subplot. The maximum PCC value is found at mobile search query data, and the minimum PCC value is found at PC search query data. The potential reason is that trading stocks on mobile devices are convenient now, which makes statistics on the mobile side more interpretable. The correlation of stock 0000001 search query in PC part keeps a low level, which suggests that the series is strongly disturbed by some unrelated factors.

It should be noted that the keywords of different companies are linked to their industrial structure. Ping An Bank is a subsidiary of a state-owned enterprise, the volatility of the search query index is also affected to some extent by the reputation of the parent company. Vanke is an independent joint-stock company, there is no external influence of the parent company, so the search query index is highly related to it. Therefore, although they are all listed companies with shareholding systems, there is a significant difference between the logarithmic search query index and logarithmic turnover values.

## 6. Conclusions

This paper focuses on predicting the upcoming stock turnover values by fusing its past observations and Baidu search query indices.

To benefit from both weekly and daily data, we proposed a DSAR model. The proposed DSAR extracts series potential patterns from the search query indices and turnover values data by exploiting two sequential processing components. Intensive experiments on two stock turnover data collections reveal the effectiveness of our proposed method. Furthermore, DSAR can play a significant effect on small-scale data with the streamlined structure in predicting future values.

In the future, the multi-horizon prediction will be further studied, and the simultaneous prediction of multiple financial sequences as well.

## Acknowledgments

This work was supported in part by the Natural Science Foundation of Fujian Province of China (Nos. 2018J01539 and 2019J01713). The authors would like to thank the editor and anonymous reviewers for their helpful comments in improving the quality of this paper. The authors are very appreciated for the great encouragement and valuable suggestions from MPAcc students in Jimei University.

## References

1. E. Diaconescu, The use of NARX neural networks to predict chaotic time series, *WSEAS Trans. Comput. Res.* **3**(3) (2008) 182–191.
2. C. H. Eunsuk Chong and F. C. Park, Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies, *Expert Syst. Appl.* **83** (2017) 187–205.
3. L. Zhang, C. Aggarwal and G.-J. Qi, Stock price prediction via discovering multi-frequency trading patterns, *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 2141–2149.
4. J. Fan, L. Xue and J. Yao, Sufficient forecasting using factor models, *J. Econ.* **201** (2017) 292–306.
5. R. Akita, A. Yoshihara, T. Matsubara and K. Uehara, Deep learning for stock prediction using numerical and textual information, *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)* (IEEE, 2016), pp. 1–6.
6. X. Zhou, Z. Pan, G. Hu, S. Tang and C. Zhao, Stock market prediction on high-frequency data using generative adversarial nets, *Math. Probl. Eng.* **2018** (2018) 1–11.
7. Y. Pan, Z. Xiao, X. Wang and D. Yang, A multiple support vector machine approach to stock index forecasting with mixed frequency sampling, *Knowl.-Based Syst.* **122** (2017) 90–102.
8. X. Zhang, Y. Zhang, S. Wang, Y. Yao, B. Fang and P. Yu, Improving stock market prediction via heterogeneous information fusion, *Knowl.-Based Syst.* **143** (2018) 236–247.
9. J. Fang, G. Gozgor, C.-K. M. Lau and Z. Lu, The impact of Baidu index sentiment on the volatility of China's stock markets, *Financ. Res. Lett.* **32** (2020) 101099.
10. A. Shamiri and A. Nor, Modeling and forecasting volatility of the Malaysian and the Singaporean stock indices using asymmetric GARCH models and nonnormal densities, *Int. J. Trade Econ. Financ.* **1** (2007) 83–102.

11. G. C. Reinsel, G. E. P. Box, G. M. Jenkins and G. M. Ljung, Time series analysis: Forecasting and control, *J. Time Ser. Anal.* **37** (2015) 712.
12. R. Oppenheim, Forecasting via the Box–Jenkins method, *J. Acad. Market. Sci.* **6** (1978) 206–221.
13. Z. Wang, Y. Huang, B. He, T. Luo, Y. Wang and Y. Lin, TDDF: HFMD outpatients prediction based on time series decomposition and heterogenous data fusion in Xiamen, China, *15th Int. Conf. Advanced Data Mining and Applications*, China, 21–23 November 2019, pp. 658–667.
14. X. Zhou, G. Zhang, J. Sun, J. Zhou, T. Wei and S. Hu, Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based heft, *Future Gener. Comput. Syst.* **93** (2019) 278–289.
15. J. Bai and P. Perron, Estimating and Testing Linear Models with Multiple Structural Changes, *Econometrica* **66** (1998) 47–78.
16. S. Tong and D. Koller, Support vector machine active learning with applications to text classification, *Journal of Machine Learning Research* (2000) 45–66.
17. B. M. Henrique, V. A. Sobreiro and H. Kimura, Stock price prediction using support vector regression on daily and up to the minute prices, *J. Financ. Data Sci.* **4** (2018) 183–201.
18. J. Friedman, Greedy function approximation: A gradient boosting machine, *Ann. Stat.* **768** (2000) 1189–1232.
19. S. Deng, C. Wang, M. Wang and Z. Sun, A gradient boosting decision tree approach for insider trading identification: An empirical model evaluation of China stock market, *Appl. Soft Comput.* **83** (2019) 105652.
20. W. Yuchi, E. Gombojav and Boldbaatar, Evaluation of random forest regression and multiple linear regression for predicting indoor fine particulate matter concentrations in a highly polluted city, *Environ. Pollut.* **245** (2019) 746–753.
21. T. Chen and C. Guestrin, XGboost: A scalable tree boosting system, *Proc. 22nd ACM SIG KDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, p. 785.
22. Z. C. Lipton, A critical review of recurrent neural networks for sequence learning, arXiv preprint, arXiv:1506.00019 (2015).
23. S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.* **9** (1997) 1735–1780.
24. K. Cho, B. van Merrienboer, D. Bahdanau and Y. Bengio, On the properties of neural machine translation: Encoder–decoder approaches, *Proc. Empirical Methods Natural Language Processing Workshop*, Doha, 2014, pp. 103–111.
25. S. Sengar and X. Liu, An efficient load forecasting in predictive control strategy using hybrid neural network, *J. Circuits Syst. Comput.* **29** (2019) 567–576.
26. Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature* **521** (2015) 436–444.
27. J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Netw.* **61** (2015) 85–117.
28. I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning* (MIT Press, 2016).
29. W. Long, Z. Lu and L. Cui, Deep learning-based feature engineering for stock price movement prediction, *Knowl.-Based Syst.* **164** (2019) 163–173.
30. A. Brabazon and M. O'Neill, *Natural Computing in Computational Finance (Volume 2): Introduction*, Natural Computing in Computational Finance (Springer, Berlin, Heidelberg, 2009), pp. 1–5.
31. E. Chattoe, *Genetic Algorithms and Genetic Programming in Computational Finance*, ed. S.-H. Chen, Vol. 7 (Springer Science & Business Media, 2012).

32. A. Ponsich, A. Lopez Jaimes and C. Coello, A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications, *IEEE Trans. Evol. Comput.* **17** (2013) 321–344.
33. R. Aguilar-Rivera, M. Valenzuela-Rendon and J. Rodriguez-Ortiz, Genetic algorithms and Darwinian approaches in financial applications: A survey, *Expert Syst. Appl.* **42** (2015) 7684–7697.
34. M. Tapia, C. Coello and C. A. Coello Coello, Applications of multi-objective evolutionary algorithms in economics and finance: A survey, *IEEE Congress Evolutionary Computation*, Vol. 10, IEEE, 2007, pp. 532–539.
35. Q. Wen *et al.*, Robust trend: A Huber loss with a combined first and second-order difference regularization for time series trend filtering, arXiv preprint arXiv:1906.03751 (2019).
36. Y. Wang, J. Gu, Z. Zhou and Z. Wang, Diarrhoea outpatient visits prediction based on time series decomposition and multi-local predictor fusion, *Knowl.-Based Syst.* **88** (2015) 12–23.
37. D. M. Juan Aranda and H. Carrillo, Multimodal wireless sensor networks for monitoring applications: A review, *J. Circuits Syst. Comput.* **29**(2) (2020) 2030003.
38. J. Cao, Z. Li and J. Li, Financial time series forecasting model based on CEEMDAN and LSTM, *Phys. A, Stat. Mech. Appl.* **519** (2019) 127–139.
39. R. Cleveland, W. Cleveland, J. McRae and I. Terpenning, STL: A seasonal-trend decomposition procedure based on loess, *J. Off. Stat.* **6** (1990) 3–33.
40. Q. Wen *et al.*, RobustSTL: A robust seasonal-trend decomposition algorithm for long time series, *33rd AAAI Conf. Artificial Intelligence*, Vol. 33, 2019, pp. 5409–5416.
41. S. Lee and D. Fambro, Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting, *Transport. Res. Rec.* **1678** (1999) 179–188.
42. Z. Yuan, R. Zhang and X. Shao, Deep and wide neural networks on multiple sets of temporal data with correlation. In *Proceedings of the 2018 International Conference on Computing and Data Engineering* (2018), pp. 39–43.
43. I. Sutskever, O. Vinyals and Q. V. Le, Sequence to sequence learning with neural networks, *Proc. Neural Information Processing Systems*, 2014, pp. 3104–3112.
44. Y. Qin *et al.*, A dual-stage attention-based recurrent neural network for time series prediction, arXiv preprint arXiv:1704.02971 (2017).
45. Y. Tao *et al.*, Hierarchical attention-based recurrent highway networks for time series prediction, arXiv preprint arXiv:1806.00685 (2018).
46. S. Shih, F. Sun and H. Lee, Temporal pattern attention for multivariate time series forecasting, *Mach. Learn.* **108** (2019) 1421–1441.
47. Y. Chang *et al.*, A memory-network based solution for multivariate time-series forecasting, arXiv preprint arXiv:1809.02105 (2018).
48. G. Lai *et al.*, Modeling long- and short-term temporal patterns with deep neural networks, *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 95–104.
49. J. Zhou, J. Sun, P. Cong, Z. Liu, T. Wei, X. Zhou and S. Hu, Security-critical energy-aware task scheduling for heterogeneous real-time MPSoC in IoT, *IEEE Trans. Serv. Comput.* **13** (2019) 745–758.
50. J. Zhou, M. Zhang, J. Sun, T. Wang, X. Zhou and S. Hu, DRHEFT: Deadline-constrained reliability-aware HEFT algorithm for real-time heterogeneous MPSoC systems, in *IEEE Trans. Reliabil.* (2020), pp. 1–12, doi: 10.1109/TR.2020.2981419.

51. J. Zhou, J. Sun, M. Zhang and Y. Ma, Dependable scheduling for real-time workflows on Cyber-Physical cloud systems, in *IEEE Trans. Ind. Inf.* (2020), doi: 10.1109/TII.2020.3011506.
52. Z. Wang, Y. Huang, B. He, T. Luo, Y. Wang, Y. Fu and C. Huang, Short-term infectious diarrhea prediction using weather and search data in Xiamen, China, *Sci. Program.* **2020** (2020) Article ID 8814222, 12 pages, 2020. https://doi.org/10.1155/2020/8814222.