Individualized Indicator for All: Stock-wise Technical Indicator Optimization with Stock Embedding

Zhige Li
Shanghai Jiao Tong University
Shanghai, China
l-zhige@outlook.com

Derek Yang
Tsinghua University
Beijing, China
dyang1206@gmail.com

Li Zhao
Microsoft
Beijing, China
lizo@microsoft.com

Jiang Bian
Microsoft
Beijing, China
jiang.bian@microsoft.com

Tao Qin
Microsoft
Beijing, China
taoqin@microsoft.com

Tie-Yan Liu
Microsoft
Beijing, China
tyliu@microsoft.com

ABSTRACT

As one of the most important investing approaches, technical analysis attempts to forecast stock movement by interpreting the inner rules from historic price and volume data. To address the vital noisy nature of financial market, generic technical analysis develops technical trading indicators, as mathematical summarization of historic price and volume data, to form up the foundation for robust and profitable investment strategies. However, an observation reveals that stocks with different properties have different affinities over technical indicators, which discloses a big challenge for the indicator-oriented stock selection and investment. To address this problem, in this paper, we design a Technical Trading Indicator Optimization(TTIO) framework that manages to optimize the original technical indicator by leveraging stock-wise properties. To obtain effective representations of stock properties, we propose a Skip-gram architecture to learn stock embedding inspired by a valuable knowledge repository formed by fund manager's collective investment behaviors. Based on the learned stock representations, TTIO further learns a re-scaling network to optimize the indicator's performance. Extensive experiments on real-world stock market data demonstrate that our method can obtain the very stock representations that are invaluable for technical indicator optimization since the optimized indicators can result in strong investing signals than original ones.

CCS CONCEPTS

• **Theory of computation → Graph algorithms analysis**; • **Applied computing → Secure online transactions**; • **Computing methodologies →** *Artificial intelligence*;

KEYWORDS

Technical analysis; trading indicator optimization; stock embedding

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330833

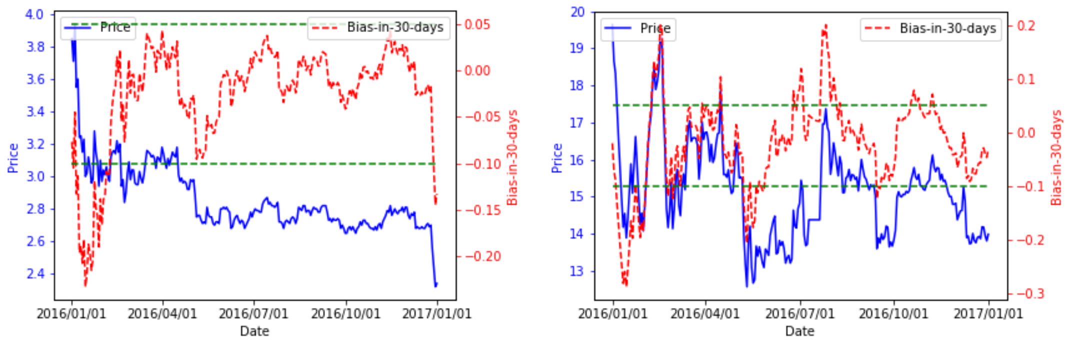
ACM Reference Format:
Zhige Li, Derek Yang, Li Zhao, Jiang Bian, Tao Qin, and Tie-Yan Liu. 2019. Individualized Indicator for All: Stock-wise Technical Indicator Optimization with Stock Embedding. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330833

1 INTRODUCTION

Technical analysis [15, 22, 24], as one of essential approaches in quantitative investment, focuses on interpreting and forecasting stock movements in terms of its price and volume. The central assumption in technical analysis lies in that all relevant information for investment decision is reflected by price and volume movement. As a result, the price and volume data constitute adequate information to make various decisions for various tasks, including market trend prediction, stock picking, and portfolio management.

In real quantitative investment, technical analysis is mainly used to select stocks targeting higher future return. In particular, to deal with the noisy nature of financial market, *technical trading indicators* are developed based on prices and volumes to provide reliable trading signals [6, 11, 19, 23], analogous to feature engineering in a general machine learning approach. More concretely, a generic technical trading indicator is usually generated based on a unified mathematical transformation over the raw price and volume data of each stock. In real world, human experts have summarized a variety of robust technical trading indicators based on their domain knowledge in financial markets. Regarded as the hand-crafted features in feature engineering, a variety of technical indicators essentially represent patterns of price and volume of diverse aspects and assess the stock moving trend in a robust and comprehensive perspective. For instance, the moving average indicator, smoothing the price sequence and discarding some randomness, is a good one to describe the trend [19, 23]. The *bias* indicator, which is computed as the deviation of the current price from its moving average, can robustly reflect the present condition of the stock price. As a result, the technical indicator, as a unified transformation over price and volume for all the stocks, plays a crucial role in various investment tasks.

However, such unified transformation yields certain limitations for the task of selecting stocks with higher future profits, as it does not consider the intrinsic properties of the stock. As a matter of fact, we observe that stocks with different properties have different



(a) An example of stable stock with ID 601558.

(b) An example of cyclical stock with ID 000892.

Figure 1: The time series data in terms of Price (blue line) and Bias-in-30-days indicator (red dash) for two specific stocks in Chinese stock market. It is clear to see that, the *bias* indicator is bounded in a certain scope for a stable stock while it ranges in a relatively large scale for a cyclical stock; for example, a *bias* value of -0.1 may be very normal for a cyclical stock, meanwhile the same value can imply a significant information for a stable stock.

affinities over indicators. Particularly, we find that even two stocks, yielding exactly the same value for one certain indicator, may stay at inherently quite different states due to the stock's own properties. For instance, Figure 1 illustrates the time series data in term of price and Bias-in-30-days indicator for two specific stocks. From these figures, we can find that, for the cyclical stock example 1b, which moves with the broader market and often with greater volatility, the *bias* indicator could easily jump to extreme value; while for the other stock example 1a, which is a strong, stable and mature, with a long history of consecutive quarterly dividends, the *bias* indicator in most cases will be limited in a certain scope. Furthermore, these figures also reveal that even if both stocks obtain the same extreme values of *bias*, the significance is totally different. The cyclical stock will be considered in its normal state so this indicator will have few effect for reference. While for the stock whose price is steady, high values of the *bias* indicators will provide rich information which is useful for both prediction and investment. This phenomenon has provided a clear indication that the unified transformation of a technical indicator is inadequate to distinguish the future profits of different stocks.

Motivated by this observation, we believe that stock-wise transformation over raw price and volume data is necessary for generating more powerful indicators. A straightforward way is to create a re-scaling factor for each individual stock, which can, however, easily lead to the over-fitted transformation. A further observation reveals that stocks with *common characteristics* have similar affinities toward the value of the indicators. For instance, most of the stocks of blue chip companies[1], whose stock price usually varies in a small scope, tends to give rise to low *bias* values in most cases. Such observation inspires us to perform the stock-wise transformation based on common stock characteristics to avoid the potential over-fitting problem. Consequently, in this paper, we propose a Technical Trading Indicator Optimization(TTIO) model that learns to optimize the original factor by re-scaling based on stock properties.

The essential challenge lies in how to get an effective representation of stock, which can reflect knowledge and belief of experienced investors. For example, effective representations should be able to distinguish steady stock from cyclical one as in above figure. One straightforward method is to collect such information manually from experienced fund manager. However, it yields quite low efficiency due to high demand of human cost with unstable accuracy as a result of human subjective instincts toward stocks.

To address these challenges, we propose to learn stock representation, i.e., stock embedding, from collective behaviors of a bunch of experienced fund managers. Because each fund manger has his own preference and specialization on stocks of various characteristics, for instance, some fund managers will prefer steady stocks to achieve the profit with low risk while some other may prefer stocks with larger volatility to obtain a greater return. Based on this fact, We make an assumption that stocks within the same fund are likely to share some common characteristics. Thus, we seek to learn embeddings for stocks with the object of preserving the such relationship between stocks, that stocks held by the same funds will obtain similar embeddings.

Given the embedding of the stocks, it is critical to design an effective approach to optimize the technical trading indicators by exploiting such representation. To this end, we propose to learn TTIO model to optimize the raw technical indicators with the information from stock embedding in order to maximize the effectiveness of the optimized indicator. Moreover, to avoid the over-fitting problem that could be caused by complex modeling, we construct the TTIO model with merely a simple one-layer neural network to optimize the raw technical indicators.

Empirically, we conduct the experiments on real-world data. We evaluate the indicator by checking its Information Coefficient(IC). In addition, we simulate the stock investment on account of the new indicators. Furthermore, we predict the rank of the profit by utilizing the optimized indicators and evaluate its effectiveness by

[1]A blue chip is a nationally recognized, well-established, and financially sound company contributed by their long record of stable and reliable growth

applying some trading strategies. Comparing with traditional indicators, the results show that our optimized indicator representation are significantly better and more stable on recent years.

In summary, the major contributions of our work include:

- We propose an indicator optimization model to achieve better indicators' performance by integrating stock-wise distinct properties.
- In order to represent stock with different properties, we learn the stock embedding based on the collective behaviors of the fund managers.
- We conduct experiments on real world stock data and evaluate the effectiveness of new indicator optimization approach by practical metrics employed in real investing strategies.

2 BACKGROUND

In this section, we will give a basic introduction on technical analysis and some technical trading indicators. We show how to apply unified transformation on price and volume to generate technical trading indicators. Furthermore, we will describe how such trading indicator can be used to construct portfolio(selecting stocks) as well as the evaluation metrics for assessing the effectiveness of indicators.

2.1 Technical Analysis

In general, there are two primary methods used to analyze stocks and make investment decisions: fundamental analysis and technical analysis. Fundamental analysis involves analyzing a company's financial statements to determine the fair value of the business, while technical analysis assumes that a stock's price already reflects all publicly-available information and instead focuses on the statistical analysis. Formally, technical analysis is the study of how past and present price action in a given financial market may help determine its future direction [15, 24, 29]. As a result, the price and volume data is comprised of all the information to make prediction in technical analysis.

2.2 Technical Indicator

In order to recognize trading patterns for specific assets based on corresponding price and volume data in technical analysis, technical trading indicators are designed in the form of the mathematical calculation on stocks' price or volume to predict market trends [1, 14].

Table 1: A set of popular technical indicators with respective calculation formulas

Technical Indicator	*Calculation Formula*
$EMA_m(i)$	$EMA_m(i) = [P_{close}(i) - EMA_m(i-1)] \times \frac{2}{m+1} + EMA_m(i-1)$
$MACD(i)$	$EMA_m(i) - EMA_n(i)$
$KLength(i)$	$P_{close}(i) - P_{open}(i)$
$KUpperLength(i)$	$P_{high}(i) - \max(P_{open}(i), P_{close}(i)))$
$KLowerLength(i)$	$\min(P_{close}(i), P_{open}(i)) - P_{low}(i)$
$Bias_m(i)$	$P_{close}(i) - \frac{1}{m}\sum_{j=0}^{j=m-1} P_{close}(i-j)$
$ROC_m(i)$	$(P_{close}(i) - P_{close}(i-m))/P_{close}(i-m)$
$MeanAmplitude_m(i)$	$\frac{1}{m}\sum_{j=0}^{j=m-1}(P_{high}(i-j) - P_{low}(i-j))/P_{close}(i-j)$

In this paper, we will study those well-recognized technical indicators that are mathematically calculated on a time series of prices and returns. Table 1 demonstrates a set of popular technical indicators with their respective calculation formulas, where in day i, $P_{high}(i)$ denotes the highest price, $P_{low}(i)$ denotes the lowest price, $P_{open}(i)$ denotes the opening price and $P_{close}(i)$ denotes the closing price. Here summarized the main purposes of the most common used technical indicators as follows:

- Exponential Moving Average(EMA) is a metric that smooths out the day-to-day fluctuations of price but more weight is given to the latest data.
- Moving Average Convergence/Divergence(MACD) is calculated by the difference between two EMA lines where one line is from a long period(slow line) and the other is from a short period(fast line). The value for m and n are usually 12 and 26.
- K Line(KL) based indicators are often applied to discover the trends and the fluctuations of the stocks. KL related indicators include K Upper Length, K Lower Length and K Length.
- Bias related indicators measure the deviation level of prices compared to simple moving average.
- ROC related indicators imply the amplitude of variation compared to previous days.
- Amplitude related indicators quantify the oscillating property in a period of recent time.

2.3 Indicator-based Portfolio Construction

As effective technical trading indicators can imply the market future trend, technical traders usually rely on them to construct portfolio for the purpose of maximizing the return[13]. A straightforward but widely-used way is to pick top k stocks ranked by the indicator's value to form up the portfolio at a certain trading time point. And, traders usually distribute investing evenly over these top k stocks, in order to reduce the risk in case the rank by indicator value cannot be perfectly consistent with that by stock's return at the certain moment. Thus, k is a hyper parameter to balance between the robustness and profits of the portfolio.

Another approach to increase the robustness of investing strategy is to leverage indicator combination. Since different indicators can capture various price patterns covering stock's different properties including trend, momentum and volatility, single indicator has its limit and may lost its efficacy in some certain circumstances. On the other hand, it is quite beneficial to combine multiple indicators together to increase the investing robustness by conquering the invalidness of single indicators. Thus, after applying certain indicator combination method, we can manage the portfolio based on the new combined indicator values and invest top k stocks with equal weight.

2.4 Evaluation Metrics for Indicator Effectiveness

In the domain of quantitative investment, information coefficient (IC) is commonly used as a performance metric for the predictive skill of a financial analyst, and also the effectiveness of the trading indicator. One of the most popular type of IC is Rank IC, which can

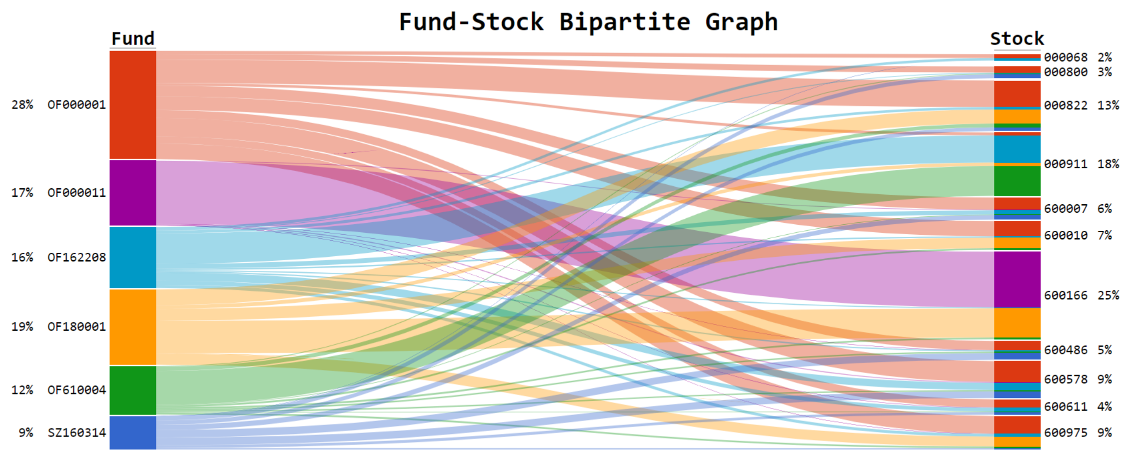


Figure 2: The general data scheme for historic portfolios of fund managers. An investment portfolio is constructed by each fund manager. Each fund can contain a list of stocks, meanwhile each stock can be held by different funds. The weight of an edge between a fund and a stock represents the distribution that the fund invests on the stock.

be calculated by : $Rank\ IC(I, R, t) = corr(order^I_{t-1}, order^R_t)$, where I means the value of the indicator, R means the profit return, $order^I_{t-1}$ means the rank of indicator at time $t-1$, $order^R_t$ means the rank of profit return at time t. $corr$ means the Pearson's product-moment coefficient[2]. Clearly, the range for IC will be between -1 and 1. Some trading indicators have positive effects on portfolio management while some may have negative effects, but both situations will be regarded as helpful for constructing portfolios.

As pointed out earlier, stocks with different properties have different affinities over the same indicator. Inspired by this, a stock-wise adjustment based on stock properties is necessary to refine existing technical indicators. We first propose a stock embedding approach to represent stock properties, then we show how to learn a stock-wise adjustment based on extracted stock embedding.

3 STOCK EMBEDDING

In this section, we aim to obtain effective representations to reflect stock properties based on the rule that stocks with similar properties have similar representations. One straightforward way is to employ human experts to conduct manual labeling. However this is quite unrealistic as a result of the high requirements of its efficiency and robustness. Instead, we seek to tackle this problem from a data mining view. Specifically, we propose to mine the stock embedding, as its latent representation, from the historic portfolios of a bunch of mutual fund managers in the market, because such data comprises a valuable knowledge repository with rich investment relation information between funds and stocks.

Specifically, We find that each fund manger has his own preference and specialization on various stocks, such as cyclical/steady stocks, high/low risk stocks. As a result, stocks within the same fund as likely to share some common characteristics. Motivated by this observation, we propose to build a fund-stock graph and adopt a random walk approach to learn similar embedding for stock nodes that are likely a co-occur in random walk.

In the following section, we will first describe the historic portfolio data of mutual fund managers. Then, we propose a bipartite graph to represent the rich relation information between funds and stocks. Based on the existing graph embedding method [4, 10, 26, 28, 36], we can obtain the stock representations with the constructed graph, where stocks with similar properties will tend to obtain the similar embeddings.

3.1 Historic Portfolios of Fund Managers

As shown in Figure 2, the real-world data in terms of historic portfolios of fund managers is usually organized as two kinds of elements, i.e., funds and stocks, and the investing relation between the. Specifically, the edge between fund f_i and stock s_j, i.e., $w(f_i, s_j)$, represents the investment of fund f_i on stock s_j. Moreover, one fund can contain a list of different stocks and one stock can be held by various of different funds.

Such historic portfolios have in fact enabled us to obtain reasonable stock property embedding. We find that each fund manager has his own preference and expertise on various stocks. For instance, some experienced fund managers may prefer to keep investing with heavier allocation on those stocks with relatively stable price sequence. As a result, these stocks are more likely to yield similar properties as they tend to be held by a set of fund managers. This exactly leads to our assumption that stocks held by the same fund are likely to share some common properties. Thus we are willing to learn similar embeddings for stocks held by the same fund.

Note that the stock properties reflecting investment preference could change over time. Yet in this section, we focus on learning a static embedding at certain time step t and we will discuss how to address the dynamic nature of the stock property in next section.

3.2 Bipartite Graph Construction

Given a bipartite graph $G = (U, V, E)$, whose vertices can be divided into two disjoint and independent sets U and V, it is quite natural to represent the funds' historic portfolio data into the bipartite graph by mapping U to the set of funds, V to the set of stocks, E to the specific investment of fund on stocks. Such constructed graph enables us to learn the embeddings for the nodes, i.e., stocks, by considering the investing relationship between funds and stocks in the form of its bipartite structure.

3.3 Learning Stock Embedding within Bipartite Fund-Stock Graph

Based on the bipartite Fund-stock Graph, We adopt an approach based on skip-gram architecture to learn embedding for the stock node where nodes appear near to each other will be likely to obtain similar embeddings. Specifically, we first generate stock-node random walk sequence based on transition probability on graph. Then we learn stock embedding that maximize the probability of neighbor node.

3.3.1 Random Walk on Fund-Stock Graph. In a random walk, the walker starts to walk from a node and at each step it moves to the neighboring node selected randomly based on the weights of the edges. Thus, in each node sequence generated by the random walk, the nodes which are connected and close to each other will have higher probabilities to appear together. Thus, We tend to generate sequence of stock-node on Fund-Stock Graph, where stocks connected by the same funds, which means that they have similar characteristics, will have higher probabilities to appear together in the random walk node sequence. Since we only care about stock embedding here, which means that we do not generate embeddings for funds, we propose a two-step random walk methods, where only nodes of stocks are selected into the sequence and optimized to get the embeddings.

Each time we start from a stock-node, and walk to next node in the following way. Suppose the node v_k indicates the kth node in the random walk sequence, then the probability of going from a stock node s_i to fund-node f_j , is calculated as follows:

$$P(v_k = f_j | v_{k-1} = s_i) = \begin{cases} \frac{w_{f_j,s_i}}{\sum_t w_{f_t,s_i}} & \text{if } (f_j, s_i) \in E \\ 0 & \text{otherwise} \end{cases}$$

Similarly, the probability of go from fund-node f_j to stock-node $s_{i'}$, is calculated as follows:

$$P(v_{k+1} = s_{i'} | v_k = f_j) = \begin{cases} \frac{w_{f_j,s_{i'}}}{\sum_t w_{f_j,s_t}} & \text{if } (f_j, s_{i'}) \in E \\ 0 & \text{otherwise} \end{cases}$$

In this way, by simply applying the fund nodes as a juncture to connect the stock nodes, we can generate a lot of sequence which only contains the stock nodes.

3.3.2 Stock Embedding by Maximizing Neighbor Node. From the generation of the stock nodes sequence, we can see that stocks which are connected by the same funds, which means that they are similar stocks based on the assumptions mentioned previously, tend to appear together in the node sequence. Thus, we propose to obtain the similar embeddings for those nodes that appear closely to each other in the random walk sequence. This problem is solved by a Skip-Gram based algorithm [20, 21], which tries maximize the probability of neighboring nodes condition on its representation give by g:

$$\max_g \sum_{u \in V} \log Pr(N_s(u) | g(u))$$

where $N_s(u)$ means the neighboring nods for the node u. Then, with the skip-gram architecture, a neural network is trained to predict the probability for each node to actually appear in the neighboring around the focus node. After that, the embeddings could be obtained through the hidden layer of the trained neural network.

4 TECHNICAL TRADING INDICATOR OPTIMIZATION MODEL

Based on learned stock embedding, we propose a new Technical Trading Indicator Optimization framework in this section. First, we briefly introduce our learning model. Second, we present our delicately-designed re-scaling model, with the purpose of keeping the original properties of the indicator in the meantime generating similar re-scaling factors for stocks with similar properties. Third, we describe our rotation learning mechanism to address the dynamic nature of the market.

4.1 Learning Model

Given an effective stock representation, how to generate a novel, better technical indicator? In this paper, we employ a machine learning approach. Instead of generating a new indicator computed by fixed mathematical equations in the traditional way, we represent the optimized and supposedly more effective indicator with a parameterized model, the neural network, and learn this model by optimizing IC, which is the performance of the indicator. In this way, an optimized indicator is generated based on the input of the stock embedding and the original technical indicator.

As discussed above, we want to achieve similar transformation for stocks with similar properties. In addition, in order to keep the property of original indicator as much as possible, we propose a one-layer re-scaling network, and the new indicator is simply a re-scaled original indicator. Through the simple design of our network, we can ensure that the stocks with similar embedding will have similar scaling score. If our model have many layers, the high non-linearity will not guarantee such characteristic.

4.2 Re-scaling Network for Indicator Optimization

Motivated by the above design principles and corresponding challenges, we propose a delicately-designed re-scaling model for indicator optimization. This model consists of two parts: a re-scaling network to generate the re-scaling weight for each indicator based on input stock embedding, and a final optimizer to generate the new indicator by multiplying the original indicator with the re-scaling weight.

4.2.1 Re-scaling Network. Our re-scaling network takes stock embedding as input, and learns to generate stock-wise re-scaling score for each indicator. More concretely, the re-scaling network contains two steps: raw re-scaling weight and weight normalization.

Raw Re-scaling Weight. We propose a simple neural network to calculate the raw re-scaling score for a specific stock i with respect to an indicator j, as shown in Eq.1, where g_i represents stock i's embedding vector, and w_j denotes parameter of the vector which needs to be optimized for indicator j. The simplicity of this equation makes it much easier to generate similar raw re-scaling score for stocks with similar representations.

$$r_{ij} = w_j^T g_i \tag{1}$$

Weight Normalization. In order to restrain the re-scaling weight into a reasonable range, we further normalize the raw re-scaling weight over all stocks via the softmax operator, as shown in Eq.2.

$$\alpha_{ij} = \frac{exp(r_{ij})}{\sum_k exp(r_{kj})} \tag{2}$$

4.2.2 Optimizing Indicator via Weighted Re-scaling. Original trading indicator can yield either positive or negative values. In order to correctly change the order for any indicator, we first normalize the raw indicator value into the range of $[0, 1]$. By multiplying the corresponding normalized indicator I_{ij} with the re-scaling weight α_{ij}, we can obtain the final optimized indicator I'_{ij}.

$$I'_{ij} = I_{ij} \cdot \alpha_{ij} \tag{3}$$

Since the new optimized indicator is expected to achieve better performance, e.g. in terms of IC, the learning target is set to maximize the metric value of correlation for gradient descent, as the Rank-IC will not generate gradient for back propagation.

$$max \;\; |corr(I'_j, R)| \tag{4}$$

In summary, the design principles aforementioned have been highlighted in this re-scaling model for indicator optimization. First, the simplicity of the whole re-scaling mechanism enables us to make proper adjustment on the raw indicator, without drastically changing its original properties. Second, the simplicity of the one layer re-scaling network makes it much easier to generate similar re-scaling weight for stock with similar properties.

4.3 Rotation Learning Mechanism

To adapt to the dynamic nature of investment, we propose a rotation learning mechanism to adjust our model's parameter over time. Belonging to online learning [3], rotation learning is a method of machine learning algorithm where data becomes available in a sequential order and is used to update the predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once.

For rotation learning, we need to decide the exact time of period for learning stock embedding T_{embed} and learning and testing technical trading indicator optimization model T_{TTIO}. The T_{embed} should be chosen long enough to effectively represent the properties of the stocks. The T_{TTIO} should be chosen neither too short nor too long considering the high volatility of price sequence and the period of validity of the optimization model. The T_{TTIO} is composed of training, validation and test process: $T_{TTIO} = T_{train} + T_{vali} + T_{test}$, where T_{test} decides the time span for rotation. We train a model for each rotation step. The algorithm is given in Alg. 1. We also adapt our stock embedding model to learn on time period T_{embed} rather than certain time step, by simply aggregating the weight for each graph in the time period T_{embed}.

Algorithm 1 Rotation Learning

```
procedure ROTATION(Embedding time period T_embed, Learning and test time period for TTIO model T_TTIO, Rotation time span t_0 = T_test. Indicator sequence {I_t}, Walk Length l, Walks per node c, Start time t_1, End time t_2)
    t = t_1
    ListNewI = []
    while t < t_2 do
        G = Construct Bipartite Graph from t − T_embed to t
        H = Stock2vec(G, l, c)
        NewI = TTIOModel(H, I_t)
        add NewI to ListNewI
        t = t + t_0
    return ListNewI
```

5 EXPERIMENT

In this section, we demonstrate the efficacy of our approach with extensive experiments. First, we present our experimental setup. Next, we apply our indicator optimization model into the real world financial market. The information coefficient is used to evaluate the performance of the optimized indicator. We compare our models with four baselines to prove the utilities of our proposed approach. In addition, we further demonstrate the effectiveness of our framework through some simple trading strategies and several case studies.

5.1 Experimental Setup

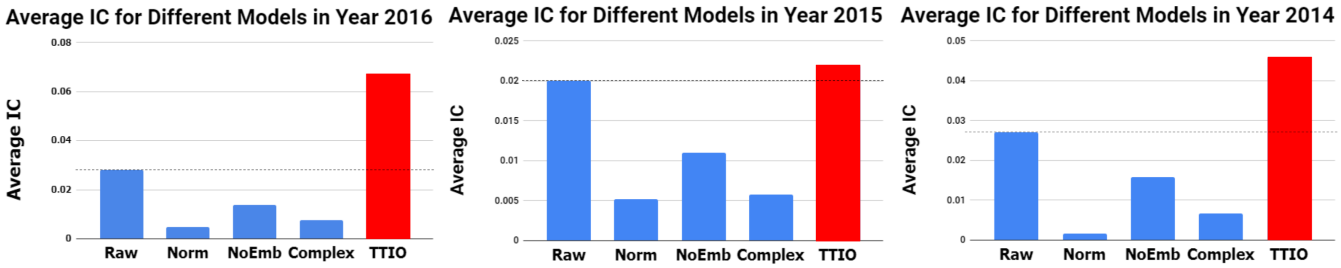
Datasets. For the indicator enhancement model, we collect the daily price and 7 classes of trading indicators from year 2013 to 2016 across over 2000 Chinese stocks, which cover the majority of the stocks in Chinese markets. Most of important indicators have been described in Table 1.

For the stock embedding model, in order to effectively reflect the stocks properties, we collect the fund manager's portfolio management in season from 2003 to 2016. All the A shares are evaluated to test the trading policies regard to actual accumulated return.

Parameter Setting. For the stocks embedding problem, we generate $c = 100$ random walks of length $l = 200$ per stock. For skip-gram architecture, the sliding window size is set to 3. The output representation for stocks have dimension $d = 32$. For the indicator optimization model, for one learning process, we split the dataset into training set(6 months), validation set(3 months) and test set(3 months). We don't use a k-folds cross validation method because we only focus on the future influence of the model.

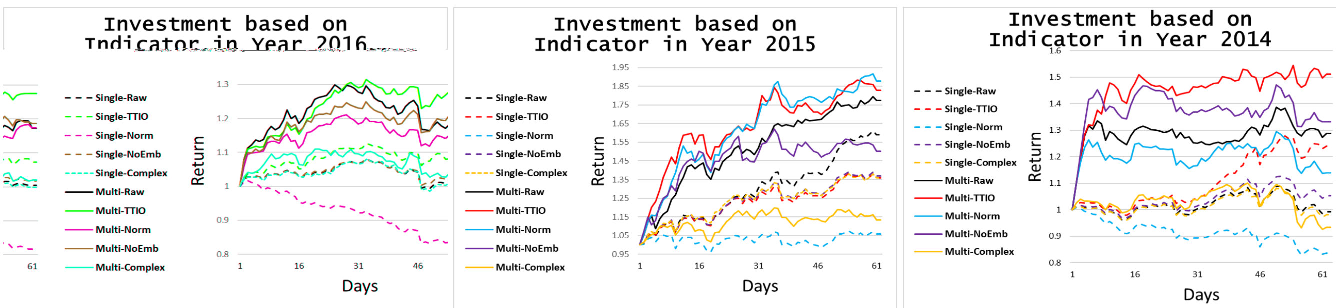
Compared methods. To evaluate the performance of our approach, we design four baselines.

- *Raw* is the raw indicators generated based on the mathematical calculation on stocks' price or volume.
- *Norm* re-scales the raw technical indicators by directly do the standard normalization calculations. Comparing with this approach will show how good enough simple scaling



(a) 2016 **(b) 2015** **(c) 2014**

Figure 3: Average Rank IC for indicators with different optimization methods in different years



(a) 2016 **(b) 2015** **(c) 2014**

Figure 4: Accumulated return generated by different indicator-driven investing strategies in different years

technique works to solve the problem of our proposed challenge.

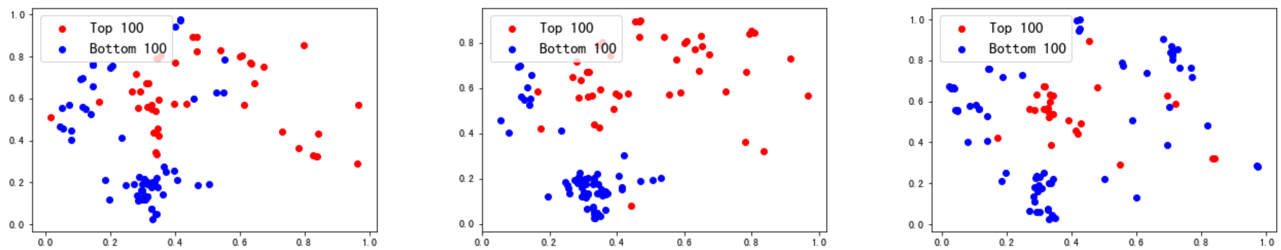
- *NoEmb* directly learns how to re-scale the technical indicators through the model without any information of the stock properties like stock embedding. Instead of calculating the scaling factors through the stock embedding, we set it as a variable optimized through back propagation. Comparing with this approach will enable us proving the effectiveness of the information aggregated through stock embedding.
- *Complex* concatenates the input of the raw indicators and the input of stock embedding and then feed it to a two-layer neural network to generate the new indicators directly from the output of the network. This approach is designed to show whether some over-fitting problems will occur on the complex networks.

Evaluation Metrics. We use Rank IC to evaluate the effectiveness of the optimized indicators. To further examine if optimized indicators can lead to increasing investment profits, we conduct experiments on indicator-driven investing. Note that, there are many options for developing a trading strategy like time-series algorithms and various machine learning techniques, which is beyond the scope of this paper. In this work, we leverage single indicator trading strategy based on the portfolio construction method described in Section 2.3, and compute the accumulated and average return on every single indicator through back-testing. And for the task of indicator combination trading strategy, we simply use a Random Forest Regression [17] to predict the rank of profits and invest on top k stocks according to the predicted value.

5.2 Experimental Results

5.2.1 Effectiveness of Optimized Indicators. We first conduct experiments to examine the performance of our indicator optimization framework with respect to various technical indicators. In particular, we use our TTIO framework to generate optimized indicator for original 7 classes of raw indicators and compute the Rank IC for both optimized and baselines. Figure 3 compares the aggregated Rank IC between optimized indicators and other comparing approaches mentioned in the previous section. From these figures, we can find that, in both 2014 and 2016, optimized indicators can outperform baselines significantly. Specifically,from the result of method *Norm* we can see that simply by doing normalization will achieve no significance in optimizing the raw indicators and even reflect almost non correlations with the profits. The similar result also occurs in approach *Complex*. The *Complex* seeks to combine the information of technical indicators and stock embedding with a multi-layer neural network and learns a new indicator one-to-one instead of scaling based on the original indicators. This method results of the over-fitting problem and also represents low IC. *NoEmb* achieves higher result than the previous two baselines. However, with the lack of the information generated from stock embedding, there is still obvious gap compared with out proposed model, which implies the effectiveness of our TTIO framework.

Figure 3 also shows that performance in 2015 is quite different from that in 2014 and 2016, after looking back to the stock market in 2015, we observe an drastically turbulent price pattern in stock market crash, which causes the severe downgrade of most of indicator groups in terms of their Rank IC.



(a) Bias (b) MACD (c) ROC

Figure 5: The embeddings for selected stocks that have maximum and minimum scaling weight leveraged by t-SNE

5.2.2 Performance in Indicator-Driven Investing. To further investigate the value of our indicator optimization framework, we conduct experiments to examine if optimized indicators can result in increasing return in indicator-driven investing. As aforementioned in Section 2.3, there are two basic indicator-driven trading strategies. One refers to evenly investing on top k stocks ranked by the value of single indicator, and the other invests top k ranked by a learned combination model, which is the Random Forest Model in this paper, over a set of indicators. In the experiment, to set the most appropriate value for k, we actually find the best k on a validation set and then apply the investing strategy with selected k to the test set. Figure 4 reports the accumulated return generated by different indicator-driven investing strategies. The investing is tested with both trading policies combining multiple indicators and trading policies only considering single indicator regard to four baselines and our proposed model.

From these figures, firstly, we can see that the year 2015, corresponding with the results of IC, shows a abnormal trading result compared with year 2014 and 2016. The method of *Norm*, which obtains the relatively bad result no matter in single trading pattern or multiple trading pattern, suddenly jumps to the best in year 2015. Generally, the investment combining multiple indicators is better than the investment only considering single indicator. In year 2014 and 2016, the results correspond with the results of indicator performance and this also proves the effectiveness of the IC for evaluation. *Norm* and *Complex* methods obtain the worst results compared to other cases. In 2014, the investment of *TTIO* with only considering the single indicator even beats the investment of *Norm* and *Complex* combining multiple indicators. Based on the result of the investment, in conclusion, except for year 2015, We can find that our model can give rise to much more accumulative return in all other cases, on both single indicator-driven trading strategy and indicator combination driven strategy .

5.2.3 Case Studies. In this experiment, we select some representative indicators and exhibit how these indicators are re-scaled on different stocks. To be more specific, we select Bias, MACD, and ROC as example indicators, and show 5 stocks that obtained the maximum and minimum raw scaling weight in TTIO model. The result is shown in Table 2. From the table we can see that different indicators show different sensitivities toward different stocks rather than simply giving heavy weights similarly. To show

	Bias		MACD		ROC	
	sector	*code*	*sector*	*code*	*sector*	*code*
max	**Public Utility**	600868	**Mechanic Engineering**	600165	**Bank**	601288
	Car	600653	**Coal**	601015	**Nonferrous Metal**	601069
	Real Estate	600747	**Building materials**	600753	**Nonferrous Metal**	600362
	Electrical Power	601558	**Transportation**	600751	**Bank**	601818
	Telecommunication	600680	**Real Estate**	600733	**Petrochemical**	601857
min	**Media**	000892	**Computer**	300379	**Building materials**	300374
	Electronic Component	300390	**Computer**	300369	**Car**	002725
	Electronic Component	300053	**Telecommunication**	300250	**Electronic Component**	000020
	Mechanic Engineering	603111	**Computer**	603019	**Telecommunication**	300211
	Medicine	300109	**Computer**	300448	**Electrical Power**	000585

Table 2: Top 5 stocks that obtained the maximum and minimum raw scaling weight on Bias, MACD and ROC

the close relation for each stocks, we first leverage t-SNE [18] to project optimized stock embedding into a two dimension. After plotting top 100 strengthened and top 100 weakened stocks into this reduced dimension, we print two different colors to these two sets, respectively. Figure 5 illustrates the colored two separate sets with respect to those three representative indicators. From these figures, we can see that our model is able to assign similar re-scaling weights to stocks with similar representations/properties, which is quite consistent with our designed principles.

6 RELATED WORK

Technical indicator is the fundamental tool in technical analysis. Previous work on indicator optimization can be roughly divided into two classes: hard-crafted indicator and indicator by deep learning.

Hand-crafted indicator has been proposed by experienced investors and economists decades ago[9, 31, 34, 37]. Gunasekarage *et al.* [11] analyzed the performance of the Simple Moving Average(SMA) indicator using index data for four emerging South Asian capital markets, and proved its predictive ability to generate excess returns. Chong and Ng [25] reported that the Relative Strength Index(RSI) as well as Moving Average Convergence-Divergence(MACD), can generate return higher than buy-and-hold strategy in most cases. Instead of considering on indicators that limit to one stock, Gatev *et al.* [8] tested a hedge fund equity trading strategy concerning the distance and the correlation of the prices. The experiments showed that trading suitably formed pairs of stocks exhibited profits, which were robust to conservative estimates of transaction costs.

With the development of deep learning, more research efforts have been paid to indicator optimization by end-to-end representation learning[16, 27, 30, 32]. These methods takes raw prices and even hand-crafted indicators as input, and learn to make trading decisions and future predictions by deep neural network. The latent representation in deep neural network can been seen as learned indicators. Takeuchi *et al.* [33] simply extract useful features without the need for extensive feature engineering, which only consider the price and the cumulative return over a certain time period for each stock example. This study represented one of the first applications of deep learning to stock trading directly utilizing prices and returns as input features. After that, various machine learning algorithms were applied to achieve some prediction tasks by directly using price and volume related features [5, 12, 35]. Furthermore, Deng *et al.* [7] introduced the contemporary deep learning into a deep reinforcement network for financial signal processing and on-line trading by feeding spread of prices into the model.

Different from existing efforts, which focus on either the hand-crafted indicator or the learning method to create new indicator, our work propose to optimize the technical indicator for stock selection by considering the stock properties. We introduce invaluable external data to learn the stock representation, and further learn to re-scale the indicator properly based on the stock representation. To the best of our knowledge, we are the first to learn technical indicator by exploiting knowledge from external data.

7 CONCLUSION

In this paper, we propose a general and explainable framework to optimize technical indicator with hidden knowledge mined from external resources. We propose a novel idea to leverage the difference in terms of indicator's stock-wise affinity and take a data mining view to learn the stock representation, by mining knowledge repository from collective behaviors of experienced investors. Then we propose a delicately-designed re-scaling network, for the purpose of retaining the original properties of the indicator and assign similar re-scaling weight to stock with similar representation. However, the indicators generated through our model do not give temporal difference thus we simply adapt it to the real world with a rough rotation learning method. So dynamically optimizing technical indicators will be left for the future work.

REFERENCES

[1] Steven B Achelis. 2001. *Technical Analysis from A to Z.* McGraw Hill New York.

[2] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer, 1–4.

[3] LÃĺon Bottou. 1998. *Online Algorithms and Stochastic Approximations.*

[4] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2016. Deep neural networks for learning graph representations. In *Thirtieth AAAI Conference on Artificial Intelligence.* 1145–1152.

[5] Rodolfo C. Cavalcante, Rodrigo C. Brasileiro, Victor L. F. Souza, Jarley P. Nobrega, and Adriano L. I. Oliveira. 2016. Computational Intelligence and Financial Markets: A Survey and Future Directions. *Expert Systems with Applications* 55, 1 (2016), 194–211.

[6] Chalothon Chootong and Ohm Sornil. 2012. Trading Signal Generation Using A Combination of Chart Patterns and Indicators. *International Journal of Computer Science Issues* 9, 6 (2012).

[7] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai. 2017. Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *IEEE Transactions on Neural Networks and Learning Systems* 28, 3 (2017), 653.

[8] Evan Gatev, William N. Goetzmann, and K. Geert Rouwenhorst. 2006. Pairs Trading: Performance of a Relative-Value Arbitrage Rule. *The Review of Financial Studies* 19, 3 (2006), 797–827.

[9] Janos Gertler and Hong Shun Chang. 1986. An instability indicator for expert control. *IEEE Control Systems Magazine* 6, 4 (1986), 14–17.

[10] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. 2016 (2016), 855–864.

[11] Abeyratna Gunasekarage and David M Power. 2001. The profitability of moving average trading rules in South Asian stock markets. *Emerging Markets Review* 2, 1 (2001), 17–33.

[12] However. 2014. Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. *Journal of Applied Mathematics,2014,(2014-3-19)* 2014, 1 (2014), 1–7.

[13] Why Read It. 1959. Portfolio selection: efficient diversification of investments. (1959).

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

[15] Charles Kirkpatrick. 2006. *Technical analysis: the complete resource for financial market technicians.* FT Press.

[16] Muneesh Kumar and Sanjay Sehgal. 2004. Company Characteristics and Common Stock Returns: The Indian Experience. *Vision* 8, 2 (2004), 33–45.

[17] Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.

[18] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[19] Massoud Metghalchi, Juri Marcucci, and Yung-Ho Chang. 2012. Are moving average trading rules profitable? Evidence from the European stock markets. *Applied Economics* 44, 12 (2012), 1539–1559.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Computer Science* (2013).

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. 26 (2013), 3111–3119.

[22] Christopher J Neely et al. 1997. Technical analysis in the foreign exchange market: a layman's guide. *Federal Reserve Bank of St. Louis Review* Sep (1997), 23–38.

[23] Christopher J Neely, David E Rapach, Jun Tu, and Guofu Zhou. 2011. Forecasting the Equity Risk Premium: The Role of Technical Indicators. *Social Science Electronic Publishing* 60, 7 (2011), 1772–1791.

[24] Christopher J Neely and Paul A Weller. 1997. Technical Analysis in the Foreign Exchange Market. *Social Science Electronic Publishing* 79, No. 2011-001 (1997), 343–373.

[25] Wing Kam Ng. 2008. Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30. *Applied Economics Letters* 15, 14 (2008), 1111–1114.

[26] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *International Conference on International Conference on Machine Learning.* 2014–2023.

[27] Jigar Patel, Sahil Shah, Priyank Thakkar, and K. Kotecha. 2015. Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications An International Journal* 42, 4 (2015), 2162–2172.

[28] Bryan Perozzi, Rami Alrfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. (2014), 701–710.

[29] Martin J Pring. 2002. *Technical analysis explained.* McGraw-Hill Companies.

[30] Qin Qin, Qing Guo Wang, Jin Li, and Shuzhi Sam Ge. 2013. Linear and Nonlinear Trading Models with Gradient Boosted Random Forests and Application to Singapore Stock Market. *Journal of Intelligent Learning Systems and Applications* 05, 1 (2013), 1–10.

[31] Alejandro RodrÃŋguez-GonzÃąlez, ÃĄngel GarcÃŋa-Crespo, Ricardo Colomo-Palacios, Fernando GuldrÃŋs Iglesias, and Juan Miguel GÃşmez-BerbÃŋs. 2011. CAST: Using neural networks to improve trading systems based on technical analysis by means of the RSI financial indicator. *Expert Systems with Applications* 38, 9 (2011), 11489–11500.

[32] Abhijit Sharang and Chetan Rao. 2015. Using machine learning for medium frequency derivative portfolio trading. *Papers* (2015).

[33] Lawrence Takeuchi and Yu-Ying Albert Lee. 2013. Applying deep learning to enhance momentum trading strategies in stocks. In *Technical Report.* Stanford University.

[34] Richard H Thaler. 2005. *Advances in behavioral finance.* Vol. 2. Princeton University Press.

[35] Jonathan L Ticknor. 2013. A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications* 40, 14 (2013), 5501–5506.

[36] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural Deep Network Embedding. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 1225–1234.

[37] Yingzi Zhu and Guofu Zhou. 2009. Technical analysis: An asset allocation perspective on the use of moving averages. *Journal of Financial Economics* 92, 3 (2009), 519–544.