Return Prediction Based on Discriminating market-styles with Reinforcement Learning

Zhiguo Bao

School of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou 450046, Henan, China

Shuyu Wang

Qianhai Zhuoshi Investment Management (Shenzhen) Co., Ltd, Haidian 100089, Beijing, China

Received: February 19, 2021. Revised: July 17, 2021. Accepted: July 28, 2021. Published: July 30, 2021.

Abstract—For hedge funds, return prediction has always been a fundamental and important problem. Usually, a good return prediction model directly determines the performance of a quantitative investment strategy. However, the performance of the model will be influenced by the market-style. Even the models trained through the same data set, their performance is different in different market-styles. Traditional methods hope to train a universal linear or nonlinear model on the data set to cope with different market-styles. However, the linear model has limited fitting ability and is insufficient to deal with hundreds of features in the hedge fund features pool. The nonlinear model has a risk to be over-fitting. Simultaneously, changes in market-style will make certain features valid or invalid, and a traditional linear or nonlinear model is not sufficient to deal with this situation. This thesis proposes a method based on Reinforcement Learning that automatically discriminates market-styles and automatically selects the model that best fits the current market-style from sub-models pre-trained with different categories of features to predict the return of stocks. Compared with the traditional method that training return prediction model directly through the full data sets, the experiment shows that the proposed method has a better performance, which has a higher Sharpe ratio and annualized return.

Keywords—Return Prediction, Quantitative Investment, market-styles, Reinforcement Learning.

I. INTRODUCTION

For mutual funds or hedge funds, return prediction has always been a fundamental and important problem. In general, a good return prediction model will directly determine the performance of a quantitative investment strategy. However, feature engineering, models, portfolio optimization, and strategies are also important for a quantitative investment strategy. Generally, The return prediction model will be the most important one.

However, the performance of the return prediction model is influenced by the market-style. Different market-styles will make some features valid, while some features will be invalid. Even models trained through the same data set will behave differently in different market-styles.

The market-style is the sum of investors' investment preferences, risk preferences, market cycles, and economic situation. It is an abstract conception. In fact, it does not currently have a precise definition. For example, when the economic situation is improving, everyone is optimistic about the market, and the market is in a rising cycle; this is a market-style. At this time, investors usually prefer small-cap stocks. Market capitalization features will be valid. On the contrary, if the economic situation is not good, everyone has pessimistic expectations of the market. The economic situation is in a downward cycle, which is also a market-style. At this time, investors usually prefer stocks with large market capitalization. This is why even if the models are trained through the same data set, they will behave differently in different market-styles.

Traditional methods have not a good solution to this problem, such as the Single Factor Model [1] and the Three Factors model [2]. They are both linear model or their variants. They want to train a general and universal model on one data set. However, the linear model has minimal fitting ability. In fact, it does not satisfy the requirements of mutual funds or hedge funds because they often have hundreds of features. At the same time, changes in market-style will make certain features valid or invalid.

Furthermore, some nonlinear models such as DA-RNN [3], and Tre-Net [4], both use complex nonlinear models based on sequence model with LSTM[5,6], combined with long-term, end-to-end raw feature data, trying to solve problems linear models and their variants meet[7-9]. However, they still do not solve the problem of changes in market-style. On the contrary, because the feature engineering is too simple, it is possible to make the model over-fitting.

A new method is proposed to solve these problems. Divide the original data set into two parts. Firstly, on the first part, classify the features according to financial domain knowledge and feature engineering. Then define the market-style with formulation, divide the sub data sets by different categories of features, and then train sub-models. Secondly, train an agent by Reinforcement Learning on the second slice of data, which automatically discriminates the market-style, and automatically selects a model that best fits it to predict return.

This method is applied to the simulated Chinese stock markets and achieves a better performance than traditional methods. The main contribution has three points: (1) Define the market-style. (2) Discriminate the market-style. (3) Predict return by discriminating market-styles.

This paper has the following structure. Section 2 introduces the proposed method, a return prediction model. To evaluate the performance, section 3 tests the models on simulated China stock markets by checking annualized return and Sharpe ratio. The last section is a conclusion and future works of this research.



Fig. 2. The process of the proposed method

II. THE PROPOSED METHOD

The proposed method mainly consists of two important parts. Divide the training data set into two parts; these two data sets do not overlap. For the first part of the data set, categorize each feature on the data set based on financial domain knowledge. Usually, there are four categories of features: the consensus features, the fundamental features, the tech features, and market microstructure features. After categorizing features, divide the data set into four sub data sets, each of which belongs to a different category. Then, train sub-models separately on these four sub-datasets. How to choose a model or an algorithm for training sub-models is also based on financial domain knowledge. For example, the fundamental features are mostly categorized features. In this case, a tree model should be used. Moreover, the market microstructure features are mostly real number features. In this case, a neural network model should be used.

On the second part of the data set, train a Reinforcement Learning agent whose task is to automatically discriminate the market-style and select a sub-model that best fits the current market-style to predict the return. Here, for the market-style of a certain day, concatenate the original financial time-series features of this day's market index and the output of each sub-model with the inputs of market index features of this day to define the market-style. This market-style vector will be the input to the Reinforcement Learning agent as a state. The agent's output is a set of probability distributions to describe the probability of selecting a sub-model pre-trained in the first part of data. Then choose a sub-model to predict the return for each stock in the market. The trading strategy will determine the final action for buy or sell based on the return prediction of each stock in the market based on this model. The reward is the sum of profits and loss of trading strategy of this day. In the end, alternately and iteratively train the agent until the return is maximized. At this time, the agent has the ability to discriminate the market-style automatically. The architecture of the proposed method is shown in Fig. 1. The process of the proposed method is shown in Fig. 2.

A. Data Partition

Different market-styles will make certain features valid or invalid. The performance of models trained through the same data set may be different at different times. For example, when the market is in a bull market, investors prefer small-cap stocks, and when the market is in a bear market, investors' preferences are not obvious. Therefore, divide the data set into two parts. On the one hand, continue to categorize the features based on financial domain knowledge and train sub-models on the sub data sets. On the other hand, in the second data set, the market-style is defined by concatenating financial time-series features of the market index this day and the output of the sub-models by the features of the market index of the day as inputs. In this way, the Reinforcement Learning agent can be trained and automatically discriminate the market-style and select the best sub-model that best fits the current market-style to predict the return. Data sets are divided into two parts; note them as D1 and D2 in Fig. 3.



Fig. 3. The data partition

B. Features Categorization

After dividing the data set into D1 and D2, firstly study the first data set D1. Use financial domain knowledge to divide the first data set into four sub data sets by columns. The consensus features data set, the fundamental features data set, the tech features data set, and the market microstructure features data set are shown in Fig. 4.

Consensus features data sets only contain consensus features derived from some brokers' research reports and analysts' consensus and expectations. For example, the company's consensus on certain key financial indicators in the coming days and its big events in the coming days, such as buy or sell ratings. They have mostly categorized features. Because such features more or less express the consensus of institutions and analysts, they are categorized as consensus data sets.

The fundamental features data set contains only fundamental features derived from the company's financial or operational aspects. For example, a typical feature is the market value, which is the company's total equity and current stock price, used to describe its total value. Another typical feature is the market value of circulation, which is the product of the company's total capital stock and the current stock price. It is used to describe the total value of the company's shares in the secondary market. The fundamental features are usually categorized features.

The tech features data set only contains tech features. These features are derived from some tech information in the stock market. They are often stock prices or their derivatives. There are usually some complicated formulas to define each tech indicator, such as the moving average. It is a typical tech feature, the average of the stock price from a certain window price. The tech features are usually real number features.

The market microstructure features data set only contains the market microstructure features derived from the exchange's market push. These messages describe the changes in the market at every moment. For example, the buyer and seller's quotes will be prioritized according to the price and time, and this information will be sent to the investors as a snapshot by the exchange by 300ms to 2s. They are often referred to as market microstructures, and changes in market microstructure are also the root cause of changes in a stock price.

Many experiments have proved that the performance of models trained through these sub-datasets is not the same. The performance of different models will also be different under different market-styles. The consensus features data sets, fundamental features data set, tech features data set, and market microstructure features data set as SD1, SD2, SD3, and SD4.



Fig. 4. Features categorization

C. Sub-Models Training

After completing the features categorization for the data set, then train sub-models on these sub data sets with different categories of features. Model selection and parameter settings are considered. For example, for subfeatures data set such as consensus and fundamental features data sets, use the Gradient Boosting Decision Tree model for training. For subfeatures data set such as tech and market microstructure, use a sequence model for training. The process is shown in Fig. 5.

(1) Gradient Boosting Decision Tree

Gradient Boosting Decision Tree[10] is a boosting model. Its core idea is to train many decision trees in each iteration and to

blend them. For the regression problem, the decision tree is a regression tree, and the purpose of this tree is to minimize loss function in the function space. For the regression task, the residual is the gradient in terms of the mean square error function. As the loss function drops, the model can be continuously improved.

The features of the consensus feature data set and the fundamental features data set are mostly category features rather than continuous real number features. Some experience and experiments have shown that the Gradient Boosting Decision Tree model can perform better on data sets with category features than the deep learning model.



Fig. 5. Sub-Models



Fig. 6. Sequence Model with one LSTM cell



Fig. 7. Market Styles and Model selector

(2) Sequence Model with One LSTM Cell

Use the sequence model on the tech features data set and market microstructure features data sets for relatively large numerical features. In linear or nonlinear models, inputs are often the features of one day. However, intuitively and experimentally, features of one day and features of the past few days determine tomorrow's return for each stock.

A very intuitive example is that if a stock price falls every day, then there is a high probability that its stock price will fall tomorrow. If a stock price rises every day, then its price will rise with a high probability tomorrow. Here, actually use the information of the past few days to predict. Use a simple sequence model with one LSTM cell to do this. Fig. 6 shows the structure of the sequence model with one LSTM cell.

D. Market-styles Definition

Market-styles Definition may be the most important and the most challenging and contributing part of the whole work. In fact, there are many options to define market-style. For example, use the average of features in a day or a few days to describe the market-style, or use all models' output in a day as the market-style.

Here, use the following method to describe the market-style. Describe and discriminate the market-style will be completed in the second data set. Use the original features of the market index of a day and outputs of the pre-trained sub-models on each feature sub data set to define the market-style. The market index' s original financial time-series features essentially describe the basic information from the consensus, fundamentals, tech, and market microstructure. The output of each sub-model contains the prediction from different feature data sets. Thus, even if one model completely loses its prediction ability today, it will not be too bad for today' s prediction because other models exist. Fig. 7 shows how to define the market-style.

This process is formalized as following:

$$S = [Xindex, Xout put]$$
(1)
Here S is market style and Xindex is the original financial

Here, *S* is market style, and *Xindex* is the original financial time-series features of a market index of a day, and *Xout put* is the output of each sub-model we trained by different categories of sub-features data set for a day.

E. Market-styles Discrimination

After completing the definition of the market-style, then discriminate the market-style through training a Reinforcement Learning agent. The process of training a Reinforcement Learning agent is shown in Fig. 8.



Fig. 8. Process of training a RL agent

Define the three important conceptions of Reinforcement Learning are state, action, and reward. Here, the state is the market style. Use the concatenating vector of the original financial time-series features of the market index of a day and outputs of the sub-models as the market style. The state is a high dimensional vector.

Action is a set of probability distributions that describes the probability of selecting a sub-model corresponding to different features data set. These models are sub-models that correspond to the different categories of features, trained on the first part of the data set.



Fig. 9. The proposed agent

In the proposed method, rewards will be defined as follows. After the agent takes action based on the state, select a model to predict stocks' return across the market. Based on the model 's predicted return for each stock, build a portfolio. In short, buy stocks whose returns are in the rank of top 50, and the bottom 50 of stocks will be sold if they hold it. Moreover, a pro fit or loss every day is used as the reward value.

F. Return Prediction

The agent gives a set of actions of the probability distribution every day by discriminating the market-style. This set of probability distributions describes the probability that selects a sub-model trained on the first part of the data set. Select a sub-model with the highest probability and then use it to predict the market's stock return. Then build a portfolio based on its predicted return. The daily profit or loss will be used as rewards. Repeat many episodes until train a correct agent that can discriminate the market-style as expect.

In the end, get an agent that can automatically discriminate the market-style, shown in Fig. 9. It can discriminate the market-style every day, select a sub-model that best fits the current market-style, and predict the return of all stocks in the market tomorrow to help build a portfolio.

III. EXPERIMENTS

In this section, demonstrate the effectiveness of the proposed method through several experiments. There is a CSI 500 data set. Use this data set separately for training, validating, and testing. Firstly, divide the data set into two parts, divide sub-features data set by financial domain knowledge, and train features sub-models. Then, define the market-style, train the market-style discriminating agent with Reinforcement Learning, and build a portfolio for simulated backtesting. Finally, evaluate the proposed methods with some indicators such as annualized return and Sharpe ratio, and compare with some traditional methods.

A. CSI 500 Data Set

In order to evaluate the overall performance of stocks of different sizes in the market. CSI Index Co., Ltd. has built a scale index system such as large, medium, small, medium-small size indexes. Providing analytic tools and performance benchmarks for the market, setting the foundation for research and development of financial products and other indices.

The stocks in the sample space of the CSI 500 data sets are composed of the top 500 stocks with the total market capitalization, excluding the CSI 300 index constituents, which are the top 300 stocks in the total market capitalization. It can represent the performance of some small and medium-sized companies in China's A-share market comprehensively.

B. Model Setting

(1) Proposed Model Setting

In the first data set, use Gradient Boosting Decision Tree for the consensus features data set and the fundamental features data set. For the tech features data set and the market microstructure features data set, use a sequence model with one LSTM cell. In the second data set, train a Reinforcement Learning agent that discriminates the market-style, where the model is a two-layer perceptron model. The architecture is shown in Fig. 1.

For the consensus features data set and the fundamental features data set, use the Gradient Boosting Decision Tree model. For the Gradient Boosting Decision Tree model, set the number of boosting rounds to 1000 rounds, the learning rate is 0.003, the max depth per tree is 7, the max number of leaf nodes is 70, and the co-sampling rate is 0.6. The early-stop rounds are 50. Not set the L1 and L2 penalty terms. In fact, experience and experiments have shown that for complex nonlinear models, the L1 and L2 penalty terms can often have a limited effect. Here, the boosting round's number of 1000 is also a parameter based on experience and experiments. In fact, many values for

boosting rounds have been tried, such as 500, 1000, 2000, 5000, etc. Finally, 1000 rounds are chosen as the hyper-parameters. Because for the hyper-parameter search tool such as hyperopt [11] or grid search, the number of rounds of 1000 rounds can often have the best performance.

For the tech feature data set and the market microstructure features data set, the model is a sequence model with one LSTM cell. The sequence model has a time series length T of 5 and a hidden unit size of 256. The output of the LSTM cell will be sent to a fully connected layer with 256 cells. The final output is a real number. Set the learning rate to 0.003, the optimizer is Adam[12], and the number of training steps is 5000. Here, set a relatively small number of training steps, because for this problem, or for a financial time series analysis problem, the training process will often converge from 5000 steps to 10000 steps, which is different from some computer vision problems that require hundreds of thousands, or millions of steps.

The market-style discriminating agent is a model based on Reinforcement Learning. Use the policy-based iterative algorithm Policy Gradient to train the model. The network structure of the agent is a two-layer perceptron model. It has two hidden layers. Each hidden layer has 256 units. Use ReLU[13] as the activation function. Its output layer is a softmax layer, which represents a set of probability distributions. Set the Reinforcement Learning model's learning rate to 0.003, the optimizer to Adam, the number of training steps to 5000, and the reward attenuation value to 0.98. It can be seen that the market-style discriminating agent model is a simple nonlinear model. It is essentially a two-layer perceptron model. The Policy Gradient algorithm only trains it.

(2) Baseline Models Setting

Two baseline models correspond to linear and nonlinear models. Not divide the data set into two parts, not divide the feature sub-datasets, not train the sub-models, not discriminate the market-style. Train a linear model, a nonlinear model, on the entire data set.

The baseline linear model is straightforward. It is actually a linear single layer perceptron without activation function. There are many ways to train this simple model, such as the least-squares method, the gradient descent method, and the orthogonal projection method. The architecture is shown in Fig. 10.



Fig. 10. The baseline linear model

The second baseline model is a sequence model with one LSTM cell. The only difference between it and the sequence model with one LSTM cell used in the tech features data set and the market microstructure features data set is that the baseline sequence model does not divide the sub data sets. It is trained directly on the entire data set.

This model has a time series length T of 5, and it has one LSTM cell. The LSTM cell has a hidden layer size of 256, and its output will be sent to a fully connected layer. The fully

connected layer has 256 units, and its output is a real number. Set the learning rate to 0.003, the optimizer to Adam, and the number of training steps to 5000. It can be seen that the hyper-parameter settings for the baseline model are expected to be as similar as possible to the model involved in the proposed method. Train this baseline sequence model by minimizing the squared error loss function by the gradient descent method. The architecture is shown in Fig. 11.



Fig. 11. The baseline nonlinear model

C. Backtesting Setting

Backtesting is a critical research process for quantitative investment. Backtesting can evaluate the historical performance of the return predicting model and trading strategy, and draw some conclusions about performance to improve the model and strategy. At the same time, backtesting can also help check whether there is the illegal use of data during the training process. For example, may accidentally use data that does not exist in historical data, that is, future data. The impact of this situation on model training and simulated trade execution is very large, because tomorrow's stock price is not known today. If the stock price of tomorrow is used today, then the prediction is meaningless, so the prediction also becomes unnecessary. It is the reason why backtesting is needed to check if there is this case. In the end, analyze the performance of models and strategies through some evaluation matrices.

The data set used in the backtesting is the CSI 500 index data set. The backtesting time setting is as follows. The training data will be from January 1, 2005, to June 30, 2015. The validation data will be from July 1, 2015, to December 31, 2015. The test period starts on January 1, 2016, and ends on December 31, 2018. 10 years of historical data is used for training, half a year of historical data for validation, and 3 years of historical data for backtesting.

For the proposed method, divide the sub data sets on the training set. The first part of the data set starts from January 1, 2005, to December 31, 2011. the second part of the data set starts from January 1, 2012, to June 30, 2015. Moreover, continue to divide the first part of the data set into four sub-features data sets. The four sub-features data sets are the consensus features data set, the fundamental features data set, the tech features data set. The sub features data sets are based on financial domain knowledge.

For each test interval, the corresponding training set and the test set will be shifted for one year. For example, for the test interval from January 1, 2016, to December 31, 2016, the corresponding training set is from January 1, 2005, to June 30, 2015, and the validation set is from 1st of January 2005 to December 31, 2015.

This setting's purpose is also designed to reduce the impact of changes of market-style on the model. A very intuitive explanation is that the market-styles of 2016, 2017, and 2018 are different in the test data set. Thus, actually, 3 models will be trained.

There are Some specific back-test parameters, such as fees and taxes. Fees and taxes are the most important source of transaction costs. In China's financial markets, this rate is about 0.002 of a transaction. In fact, if a model is trained and a trading strategy is designed to make the profit greater than 0.003 per transaction, then the model and strategy are very successful.

The back-testing setting sets the sum of transaction fees and taxes rate to 0.002 per transaction, regardless of direction. This means that whether a transaction direction is buying or selling, its transaction fee is 0.002 of the transaction amount.

D. Trading Strategy

This section introduces how a model participates in the back-testing after training. Train the return prediction model. The return prediction model's task is to predict each stock's return in the financial market for a day. Build a portfolio through the returns given by the model. The way to build a portfolio is trading strategy. A simple strategy is that sort the returns of the model predictions every day, always buy stocks that are ranked first and sell those stocks that are sorted behind.

Due to changes in market-style and the existence of fees and taxes, it cannot rely entirely on the model's predictions. Because of the transaction cost, a trading strategy will maximize the return during a certain backtesting period. On the other hand, transaction costs need to be minimized. This puts high demands on the trading strategy.

Here introduce an improved trading strategy. On the one hand, this trading strategy uses the model's return prediction for each stock. On the other hand, this trading strategy takes the transaction costs into account and does not aggressively adjust the positions. This strategy is called the Top K - Margin strategy.

Sort the model's predicted returns for each stock every day. After sorting the predicted returns, get a table of stocks and returns. Moreover, for the strategy to hold positions for each stock every day, have a table that describes the relationship between stocks and positions. Only buy the top K stocks.

A margin is a real number, usually greater than K but less than M, which is the number of shares that can be traded that day. Those whose predicted return is outside of K but is ranked within M will not be bought or sold. If these stocks are in positions, then nothing will be done. For those outside the M, if these stocks are in positions, then they will be sold.

Due to the rapid changes in market-style, one category of features may be valid or invalid, and the corresponding model may be valid or invalid. At this time, the prediction of the model will not be very reliable. So, with Margin, can avoid the model shocks caused by the rapid changes in market-style. For example, the model's return prediction for some stocks is completely the opposite for a few days. However, at this time, it cannot buy or sell this stock aggressively. For example, changes in market-style, or the loss of training data points, or an accidental breaking news event, may cause the model to predict anomalies on a certain day. By introducing Margin, the model is buffered to do not aggressively or conservatively use the model's return prediction to build a portfolio. In the strategy, the K value is 50, and Margin is 200. For the index members of the CSI 500 Index from 2005 to 2018, the number of members is 1,364. The above is the trading strategy.

E. Performance Metrics

This section introduces some indicators for evaluating model and strategy performance. Some of these indicators describe the profitability of models and strategies, and some describe the stability of models and strategies.

The first indicator is the annualized return, which is a critical indicator. The annualized return directly describes the profitability of models and strategies, is one of the most important indicators in the actual trading environment. In a literal sense, the annualized return describes the return that a model can bring in a natural year, deducted from fees and taxes. Because of the deduction of fees and taxes, this indicator has become pure. Its calculation formula is as follows:

Annualized Rturn =
$$(\frac{V_T}{V_0} - 1) \times \frac{TradingDaysaYear}{TotalTradingDays}$$
 (2)

 V_T is the asset's net value on the last trading day, and V_0 is the net value of the asset on the first trading day. The annualized return describes the change in the net asset value and reflects this change in a natural year. It can naturally conclude that the larger the annualized return, the better the model and strategy. Of course, this conclusion is relatively one-sided. In fact, this indicator does not describe the volatility of the portfolio, and volatility is also a critical indicator. It is well understood that investors all like a portfolio with large returns and small volatility on assets.

The second important indicator is the Sharpe ratio. The Sharpe ratio describes the profitability and stability of a portfolio. Alternatively, in other words, the Sharpe ratio describes the ratio of the return and volatility of a portfolio. Investors all want a portfolio with a greater return and less volatility. The former is well understood; investors want a portfolio to have better profitability, then it should have a higher return. The latter, investors hope that a portfolio has less volatility because most investors are risk-averse.

The target of the Sharpe ratio is to measure the ratio of annualized returns to annualized portfolio volatility. The formula for calculating the Sharpe ratio is as follows:

$$SharpeRatio = \frac{R_p - R_f}{\delta_p}$$
(3)

The numerator in the formula is the difference between the model's annualized return and the risk-free rate. A risk-free rate usually refers to the annualized return of the baseline. A risk-free rate is the annualized return of a baseline portfolio, not free of risk. In fact, the baseline portfolio can be an open-ended index fund, a large bank deposit certificate, a short-term or medium-term national bond, or a set of investment portfolios. In the experiments, the risk-free rate is the annualized return of the market index.

The denominator in the formula is the standard deviation of the model's daily return. Models and strategies generate transactions every day, and transactions bring profits and losses. profits and losses are called PnL. To calculate the PnL of all trading days, get a set of PnL, calculate its standard deviation, and get the denominator part of the Sharpe ratio. The standard deviation is the square root of the variance, and its unit is the same to return. Naturally, this standard deviation is not want to be large. Because the large standard deviation describes large volatility, investors do not want a portfolio with large volatility.

F. Experiment Results

Here introduce and analyze the backtesting results and experimental performance. The proposed method has obtained such results on CSI 500 data set according to the backtesting parameters and trading parameters.

During the test period, from January 1, 2016, to December 31, 2018, the proposed method has an annualized return of 18% and a Sharpe ratio of 2.23, which is an outstanding result. The linear model achieved a 4% annualized return and a Sharpe ratio of 0.77. The nonlinear model achieved a 13% annualized return and a Sharpe ratio of 1.98. It can be seen that the proposed method is better than the two baseline models. Annualized return and Sharpe ratio for each model are summarized in Table 1.

Table 1 Results of Annualized Return and Sharpe Ratio

	Annualized	Sharpe
	Return	Ratio
Proposed Model	18%	2.23
Baseline Linear Model	4%	0.77
Baseline Nonlinear Model	13%	1.98

IV. DISCUSSION

Here discuss the reasons for experiment results. Changes in market-style make certain features valid or invalid. Because of this problem, the traditional models cannot discriminate the market-style, and a category of a feature may have the opposite label in different market-styles. The linear model is completely unable to cope with this situation. Moreover, for traditional nonlinear models, this situation cannot be dealt with either.

The proposed method takes this into account. Divide the data set into two parts. At the same time, categorize features into four sub-features data sets based on financial domain knowledge. After that, train sub-models through the sub data sets. Then, train a Reinforcement Learning based agent on the second part of the data, which will select a sub-model that best fits the current market-style based on the market-style.

Through the proposed method, the model and strategy as a whole part have the ability to discriminate the market-style. Simultaneously, by discriminating the market-style, it can select a return prediction model that is most suitable for the current market-style. This makes the models and strategies more robust. Even in the face of extreme market-styles and market conditions, the market-style discriminating model can select the model that best fits this situation. However, traditional linear and nonlinear models cannot handle this situation.

There are three main contributions to the proposed approach: 1. Solve the problem of insufficient fitting ability of traditional linear models. 2. Solve the problem that a single traditional nonlinear model is easy to be over-fitting without feature engineering. 3. Solve the problem that the traditional model fails due to system time-varying.

V. CONCLUSIONS

Return prediction is a fundamental and important issue for mutual funds and hedge funds. Because the market-style of financial markets will change frequently, this will make some features valid or invalid. Moreover, this phenomenon will seriously affect the performance of the trained models. To solve this problem, propose a method that automatically discriminates the market-style and selects the return prediction model that best fits the current market-style to improve the existing methods.

Firstly, divide the training data set into two parts. In the first part of the data set categorizes the features according to financial domain knowledge and divides the four categories of features sub data sets. They are the consensus features data set, the fundamental features data set, the tech features data set, and the market microstructure features data set. Then, use four sub-features data sets to train four sub-models, all of which are return prediction models. For the consensus features data set and the fundamental features data set, use the Gradient Boosting decision tree model. For the tech features data set and the market microstructure features data set, use a sequence model with one LSTM cell.

The market-style is defined in the second part of the data set. Use the original market index financial time-series features of a day and the return prediction of all models to the market index of a day as the market-style. Then, train an agent based on Reinforcement Learning, whose task is to discriminate the market-style and select a return prediction model trained using the first part of the data set. The agent's state is the current market-style, and the agent's action is a set of probability distributions, describing the probability that selects a sub-model.

The backtesting based on the market-style discriminating agent and the pre-trained sub-models is performed. Use this method to get predicted returns, and use the Top K - Margin strategy to build a portfolio, and validate the models and strategies on the CSI 500 data set, along with traditional linear models and traditional nonlinear models. The experimental results show that the proposed method achieves better backtesting performance than the two traditional methods. Analyzed the reasons, and believe that the proposed model can discriminate the market-style. Compared with the traditional models that cannot discriminate the market-style, the proposed return prediction model can perform well because it adapts to different market-styles.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China (No.61602153, No.61702161, No.31700858),

Scientific and Technological Project of Henan Province (No.152300410207, No.162102210274, No.172102210171, No.182102210020, No.182102210213), Key Research Fund for Higher Education of Henan Province (No.18A520003).

REFERENCES

- S. Delikouras, and A. Kostakis, "A Single-Factor Consumption-Based Asset Pricing Model", *Journal of Financial and Quantitative Analysis*, vol. 54, no.1, pp. 1-15, 2018.
- [2] E. Maier, "Serial product evaluations online: A three-factor model of leadership, fluency and tedium during product search", *International Journal of Research in Marketing*, vol. 36, no.4, pp. 558-579, 2019.
- [3] M. T. Nguyen, D. H. Le, T.Nakajima, M. Yoshimi, and N. Thoai, "Attention-Based Neural Network: A Novel Approach for Predicting the Popularity of Online Content," in Proc. 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 2019, pp. 329-336.
- [4] T. Lin, T. Guo, and K. Aberer, "Hybrid Neural Networks for Learning the Trend in Time Series," in *Proc. Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017, pp. 2273-2279.
- [5] A. Sagheer, and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks", *Neurocomputing*, vol.323, no. 5, pp. 203-213, 2019.
- [6] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network", *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841-851, 2018.
- [7] Hitesh Shah, M.Gopal, "A reinforcement learning algorithm with evolving fuzzy neural networks", *International Journal of Electrical Engineering and Computer Science (EEACS)*, vol. 2, pp. 68-72, 2020.
- [8] Sussy Bayona-Oré, Rino Cerna, Eduardo Tirado Hinojoza, "Machine Learning Applied to Price Prediction in Agricultural Products", WSEAS Transactions on Business and Economics, ISSN / E-ISSN: 1109-9526 / 2224-2899, vol. 18, pp. 969-977, 2021.
- [9] Mochammad Chabachib, Renny Hapsari Kusmaningrum, H. Hersugondo, Imang Dapit Pamungkas, "Financial Distress Prediction in Indonesia", WSEAS Transactions on Business and Economics, ISSN / E-ISSN: 1109-9526 / 2224-2899, vol. 16, pp. 251-260, 2019.
- [10] X. Lei, and Z. Fang, "GBDTCDA: Predicting circRNA-disease Associations Based on Gradient Boosting Decision Tree with Multiple Biological Data Fusion", *International Journal of Biological Sciences*, vol. 15, no. 13, pp. 2911-2924, 2019.
- [11] S. S. Sandha, M. Aggarwal, I. Fedorov, and M. Srivastava, "MANGO: A Python Library for Parallel Hyperparameter Tuning," in *Proc. ICASSP* 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 3987-3991.
- [12] Z. Chang, Y. Zhang, and W. Chen, "Effective Adam-Optimized LSTM Neural Network for Electricity Price Forecasting," in *Proc. 2018 IEEE* 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2018, pp. 245-248.
- [13] K. Eckle, and J. Schmidt-Hieber, "A comparison of deep networks with ReLU activation function and linear spline-type methods -ScienceDirect", *Neural Networks*, vol. 110, no. 2, pp. 232-242, 2019.

Creative Commons Attribution License 4.0 (Attribution 4.0 International , CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 https://creativecommons.org/licenses/by/4.0/deed.en US