High-Frequency Trading Factor Mining Using Genetic Programming in the A-Share Market

ZHIYUAN CHEN

in the

Department of Electrical and Electronic Engineering

Supervised by

Dr. Liang Huang

Content

Abstract

This research presents an innovative high-frequency factor mining method based on a genetic programming to solve the problem of choosing the trading time in T+0 trading activity in the Chinese stock market. Starting from the underlying logic, this paper discusses why genetic programming are particularly suitable for high-frequency quantitative trading factor mining. Using genetic programming, this research successfully found a factor ideal for high-frequency trading, called the big wave factor. It can evaluate the possibility of significant fluctuations in stock prices in the future. Given that this factor cannot judge the direction of fluctuations, this paper customizes a feasible trading strategy for this factor. Since China's stock market does not yet support T+0 trading, this trading strategy requires holding a position in the stock in advance, which will affect the calculation of the total return. Therefore, this research proposes a new rate calculation system, especially for intraday trading, which is used to compare with the traditional moving average system strategy. In the experiment, the strategy of this paper has obtained more considerable profits than a traditional strategy.

Keywords:

High-frequency trading, Quantitative trading, Factor mining, Genetic programming

Chapter 1: Introduction

1.1 Motivation

The stock market is an open market where various company stocks are traded at agreed prices. The supply-and-demand relationship in the stock market determines the price of stocks. Since the market contains transactions between two investors, it is also called the secondary market. Investors can use a variety of trading modes in the stock market. Trading methods can be classified into value investment, trend trading, and high-frequency trading. Each trading method has its own merits. The value investment has a relatively stable rate of return, and it is not easily affected by market sentiment fluctuations. Trend trading can obtain very high yields with precise timing. With the popularization of computers, high-frequency trading gradually caught the public's attention. High-frequency trading is a trading mode that buys and sells on the same day. According to research, high-frequency trading can obtain many additional benefits that other trading methods cannot obtain [1] [2]. In addition, high-frequency trading is of great significance in improving the market. Although the a-share market has not allowed the T+0 trading mode, due to the positive impact of high-frequency trading on the market environment, the opening of markets to the T+0 trading model has become a historical trend.

Quantitative trading refers to establishing a model that takes market transaction data as input, outputs a value that reflects specific market characteristics and uses this value as a trading decision-making method. Quantitative trading is all around us, moving average system indicators and energy tide indicators are just some examples of quantitative models. Compared with the non-quantitative trading mode, quantitative trading can accurately reflect market conditions with numerical values, which are not easily affected by human emotions. However, the current quantitative trading model refers more specifically to those trading models that use output value of the model to allow a computer to place orders after the quantitative model is established automatically. This method avoids the influence of human emotions to a greater extent. Its labor cost is meagre, and the transaction speed is extremely rapid. In many financially developed countries, quantitative transactions that automatically place orders by computers have already accounted for most markets.

There are many ways to develop quantitative trading models, and some are constructed manually. Manually created models usually start from logic and use statistics and probability methods to generate models for predicting future market trends. The manual process is logically rigorous, and the model is highly interpretable. Still, the human brain’s computing power is limited, manual construction costs are extremely high, and it is difficult to immediately develop a new model for iteration when the market-style changes. In recent years, due to the technological development of artificial intelligence, more and more quantitative model builders have started to replace artificial models with models constructed by artificial intelligence algorithms. Although the model of the artificial intelligence algorithm is poor in interpretability, the iteration speed of the model is fast. An artificial intelligence algorithm can generate a large number of models without too much labor cost and time. The breakneck iteration speed allows it to adapt to different market styles.

Using artificial intelligence algorithms to build trading models usually has two primary directions: neural networks and genetic programming. Many researchers have produced good results in high-frequency trading using neural networks. However, as mentioned before, the major problem with models built by artificial intelligence algorithms is that the models are difficult to explain. A neural network-based model is almost entirely a black box and cannot be explained. Although a genetic programming model may still be hard to explain, it is much easier to explain than neural network algorithms. In addition, according to the survey, it seems that few researchers use genetic programming in the creation of high-frequency trading models. High-frequency trading is very close to the underlying logic of the transaction's supply and demand relationship. If genetic programming are used, they may produce universal trading models. Therefore, the research objective of this project is the application of genetic programming in high-frequency quantitative trading.

1.2 Introduction of genetic programming

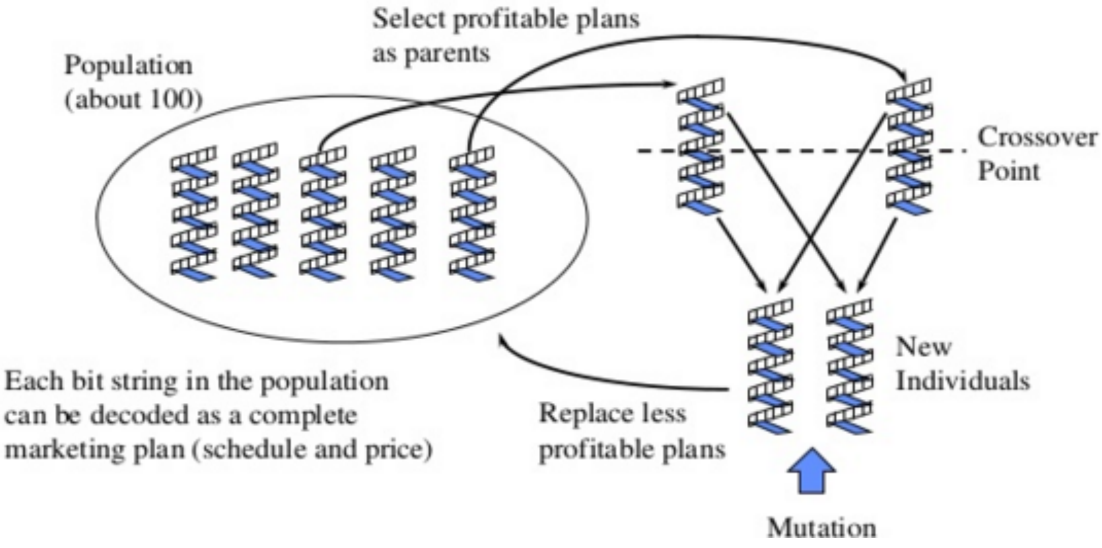

Select profitable plans as parents
Population (about 100)
Crossover Point
New Individuals
Each bit string in the population can be decoded as a complete marketing plan (schedule and price)
Replace less profitable plans
Mutation

Figure 1.1 Schematic diagram of genetic programming

Genetic programming are an example of heuristic formula evolution technology. Genetic programming simulate the process of gene evolution in nature to generate function groups that fit specific goals gradually. As a supervised learning method, the genetic programming can identify hidden mathematical functions that are difficult to discern through human brain according to specific goals. In the beginning, a set of unselected and evolved primitive functions will be randomly generated (the first-generation formula), the fitness of each part is calculated through the fitness equation, and suitable individuals are selected as the parents of the next generation of evolution. These selected parents evolve through various methods to form different offspring functions and then cycle around for the next round of development. As the number of iterations increases, operations multiply, mutate, and evolve, thus constantly approaching the truth.

In quantitative trading, factor mining is always the core technology. In previous factor research, people generally started with rules and investment experience to mine and improve the factors - that is, the method of “from Logic to Function”; Common factors such as roe, PE, PB, etc. are all researched through this method. With an increase in available market data and the development of advanced technologies such as artificial intelligence and high computing power CPU, we can use genetic planning methods to explore massive data banks, obtain some tested and effective stock selection factors through “evolution” methods, and then attempt to explain the connotation of these factors, that is, the “from Function to Logic” method. The above two methods correspond to the “deductive” and “inductive” research methods of stock factors, and both have a specific basis for existence. The latter’s advantage is that it can make full use of the considerable computing power of computers to perform heuristic searches, breaking through the limitations of human thinking, mining some hidden factors that are difficult to construct through the human brain and providing more possibilities for factor research.

1.3 Objective and Aims

After conducting relevant investigation and many experiments, this research found that models directly predicting future prices fail to achieve good returns. Given that a number of factors can affect the price, it is difficult to provide a complete input to the model. After analyzing the logic, it can be found that the transaction does not need to predict the actual stock price in the future, as long as the price fluctuates wildly. Profit can be obtained from this fluctuation. Therefore, this research aims to use genetic programming to mine factors to detect when the stock price will fluctuate significantly, called big wave factor. The main structure for this research can be summarized as follows:

- Chapter 3 preprocesses the required data and builds a genetic programming model for building big wave factor.
- Chapter 4 trains the genetic programming model, designs an appropriate trading strategy for the big wave factor, and compares it with the MACD strategy of the moving average system.
- Chapter 5 summarizes the advantages and disadvantages of the big wave factor and proposes several possible improvements.

Chapter 2: Literature Review

2.1 Trading methods

Trading methods can be divided into value investment, trend trading, and high-frequency trading in stock trading. The original foundation of **value investing** was established by Graham and Dodd [3] in their book “Securities Analysis”. The core of matter investing is to evaluate the value of a company. Price-to-earnings ratio (P/E) is a vital valuation method. Basu found that stocks with low price-to-earnings ratios outperformed high price-earnings ratios [4]. Stattman found that stocks with low price-to-earnings ratios produce excellent positive returns in the long run [5], and Rosenberg, Reid, and Lanstein [6] reached similar conclusions. Buffett is recognized as one of the best investors in value investing. Before buying stocks, he will conduct a lot of research and evaluation on the company, confirm that the company’s value is higher than the current price, and then buy a lot. From 1956 to 1969, Buffett’s investment grew at an average annual rate of over 30% of enormous compound interest, while the middle normal market is only 7~11% [7].

The advantage of value investment is that short-term speculation in the market will not affect its final profit, and its returns will be more stable. If the stock price rises, investors can get all the benefits. Since the number of transactions in value investment is less, the transaction costs generated by commissions and stamp duties will be lower than other trading methods. However, the amount of funds required for value investment is relatively high. When the target stock price falls, it is necessary to cover the position to reduce the cost.

Moreover, value investment requires higher investment and research capabilities from investors. If investment research makes mistakes, long-term holding of stocks will only lead to increasingly great losses. Even the best value investment practitioners, Buffett, makes mistakes. For example, he suffered many losses after the 2010s, and his investment in IBM even lost more than 34%.

However, Buffett's value investment may have been successful because he is in the US market. In the a-share market, the Shanghai Composite Index was 2737 points in January 2016, and only 3483 points in January 2021, and the five-year return was only 27.3%. The US stock market's Nasdaq index was 4,613 points in January 2016 and 13,070 in January 2021, with a five-year return of 183.3%. It is not difficult to see that due to differences in primary national conditions, specific asset portfolio strategies and portfolio management strategies available in the US market may not be suitable for the Chinese market.

Trend trading uses some indicators to determine the trend of stocks, buying stocks when there is an upward trend and selling stocks when there is a downward trend. Trend trading emphasizes observing the trend rather than predicting the future [8]. The logic of trend trading is that investors believe that the rise and fall of stock prices are not entirely random and will not fall immediately when there is an upward trend [9]. Investors usually refer to the indicator of stock trends as momentum. The most straightforward momentum indicator is the second derivative of the price. When the second derivative of the cost is positive, the price accelerates or slows down, and stakeholders can consider buying.

There is a factual basis for the existence of trends in stock prices. Taking the process of stock price rises as an example, when the company is optimistic, a small group of people will know the news in advance, and their purchase will cause the stock price to rise slightly. As the word spreads, more people buy, and the price rises faster. When the price is close to the expectation, a small group of people will sell first, and the price will increase and slow down. When most people believed that the price exceeded their expectations and sold the stock, the price began to fall faster. The trend effect is considered to be widespread in the investment market. Rouwenhorst found that market investors are more likely to use momentum strategies to trade and are keener to invest in stocks that have performed better in the past [10]. Between 1965 and 1995, Lee and Swaminathan used momentum effects to develop trading strategies and obtained impressive returns [11]. The basic logic used in the famous turtle trade strategy is also trend trading [12].

Compared with value investment, trend trading is more flexible. Investors who can accurately grasp the value trend can receive higher returns than value investment. According to the authoritative US investment trading magazine “Futures Truth Magazine” released in the fourth quarter of 2011, the average annual yield of the top three trend trading institutions is above 200%. The disadvantage of midline trading is that it is difficult to grasp the trend, and it is difficult to identify the right buying and selling points. The DHS model proposed by Daniel, Hirshleifer and Subrahmanyam argues that when investors receive information that can affect stock prices, they may commit two kinds of deviations. One is overconfidence: under the influence of overconfidence, investors have pushed the stock price far away from the intrinsic value, resulting in overreaction and abnormal price fluctuations. The other is underconfidence. Under the influence of underconfidence, if the private information is the same as the public and fully meets expectations, self-confidence will be significantly enhanced. If personal information is at odds with public ownership, it will not be taken seriously, and investors will choose to ignore. When the event’s occurrence is consistent with a investor’s behavior, the investor is full of confidence. When the event is inconsistent with a investor’s behavior, the investor thinks it is all external noise [13]. These two kinds of deviations lead to the same event that may induce investors to make completely different decisions, making the value trend difficult to grasp amidst the market noise.

The characteristic of **high-frequency trading** is the short holding period of the investment portfolio. High-frequency trading began in the 1930s. It used the construction of high-speed cables across the Atlantic to obtain the difference in quotations between the two exchanges for arbitrage [14]. With the development of computer technology and electronic trading, computers are designed to process large amounts of information and high-speed automated transactions, and computer-based high-frequency trading has developed rapidly. Nowadays, high-frequency trading is usually accompanied by quantitative trading models. Most high-frequency trading strategies take advantage of minor deviations from the market equilibrium.

Research shows that high-frequency trading can help improve market quality [1] [2]. The connotation of market quality is multi-level. Regarding the impact of high-frequency trading on the quality of the securities market, current research mainly examines it from two levels of liquidity and stability.

If the transaction targets on the market have good liquidity, it will show that the bid-ask spread of the entire market (Bid-ask Spread) is small, and the transaction cost will be lower. The market offers higher efficiency, so liquidity is one of the essential indicators to characterize the quality of the securities market. Since high-frequency trading can create a large volume of orders and transactions per unit of time, and both buyers and sellers generate these orders at the same time, this helps traders in the market detect counterparties more quickly, reducing the spread and matching time will ultimately reduce conversion costs [1]. Jarnecic and Snape conducted an empirical test using data from the London Stock Exchange in the United Kingdom as a sample; the participation of high-frequency trading alleviated short-term liquidity imbalances and mismatches through a series of limit orders, thereby providing liquidity for the market [2].

The meaning of market stability includes the volatility of stock prices in general and the possibility of a “flash crash” and abnormal changes in the market, that is, vulnerability. Part of the research focuses on whether high-frequency trading brings vulnerabilities to the market. This part of research focuses on “flash crash” events. The primary theoretical and empirical studies hold that the introduction of high-frequency trading has not harmed market stability ; instead, it reduced short-term market volatility. In addition, recent studies have also reached the same conclusion. For example, Brogaard et al. [15] found that high-frequency trading reduced market volatility in daily transactions and reduced market volatility during the global financial crisis in 2008. Therefore, the introduction of high-frequency trading has at least not affected the quality of the market from the perspective of volatility.

All in all, high-frequency trading has a positive effect on improving the quality of the market. The a-share market has not yet allowed T+0 trading. However, as China’s internationalization, financial openness, financial market construction, and market internationalization requires further development, high-frequency trading will likely enter the Chinese securities market in due course.

After the investment model is stable, high-frequency trading has the characteristics of lower risk and stable returns. High-frequency trading is in micro-scenarios, which means that stock price fluctuations are closer to the underlying logic. Once the model is established, there will be relatively few subsequent changes. On the other hand, for institutional investors, high-frequency trading can reduce market friction, effectively reduce the cost of shocks, and enable the entire transaction to be completed at an optimal price. For large-value transactions, high-frequency trading is an excellent way to conceal their trading behavior to prevent others from following suit. The counterparty can only see the continuous increase in trading volume but cannot know whether they are buying or selling stocks in large quantities.

2.2 Quantitative trading

According to the different investment basis, the trading mode of stocks can be divided into active trading and quantitative trading. The investment basis of active transactions is some non-data information, such as the degree of favorability of the company, the recognition of the company's statements, insider information, friend's recommendations, and even intuition, which brings great uncertainty to the transaction. It is difficult for investors to judge the accuracy of the news and the certainty of the stock price rise, and investors' trading behavior will be affected by the investor's sentiment. Quantitative trading, in a broad sense, refers to using open market information to establish mathematical models and use the model's output to make investment decisions. The commonly used indicators in the stock market, such as moving averages, RSI, OBV, etc., are all types of quantitative models. Nowadays, quantitative trading mainly refers to a trading model that uses a pre-set computer program to place orders automatically.

Quantitative investment has experienced over 30 years of overseas development, with stable investment performance. Its market scale continues to expand and is recognized by more investors [16]. In 1971, Barclays International Investment Management Corporation issued the world's first passive quantitative fund, marking the beginning of quantitative investment [17]. With the help of computer technology, quantitative traders are becoming more effective in summarizing information and placing orders. Quantitative traders can effectively reduce irrational trading decisions through trading procedures instead of subjective judgments. So far, quantitative funds have become the mainstream for securities investment. In the United States, it started to rise in 2000 and accounted for 25% of the investment market in 2005. By 2009, it had tripled, accounting for 75% of the investment market [18].

The profitability of quantitative trading should not be underestimated. Medallion, operated by Simmons, achieved an average annual rate of return of 66% in the 20 years from 1989 to 2009, which is nearly 10% higher than that of financial giant Soros and stock god Buffett and is 20% higher than the S&P 500 index over the same period. Even when the global subprime mortgage crisis broke out in 2008, the fund's return was alarming at around 82.4% [19].

The typical process of quantitative trading consists of the following three parts.

1. **Obtain data**. There are generally two types of data: time series and cross-sectional data. They usually come from data providers, and some of the data come from local databases. In most cases, they require preprocessing operations such as data cleaning.

2. **Model development**. The model is the key to the quantification of the trading model. The model quantifies the logic behind the data. Usually, the input of the model is the preprocessed data obtained in the previous processing, and the output of the model represents the winning rate of the investment. Models can be developed in many ways. The model is usually established in the early quantitative trading by the human brain [20]. A person with rich experience quantifies his investment methods to form a model. Nowadays, quantitative trading has benefited from the gradual increase in computing power and relies on artificial intelligence algorithms to build models. Standard algorithms include deep learning, reinforcement learning, and genetic programming.

3. **Backtesting**. Backtesting is a method of evaluating the model's profitability by taking historical data as the input of the model. It is the key to determining the validity of the model.

2.3 Algorithm to build the quantitative model

Quantitative trading usually uses quantitative models to make trading decisions. The methods of constructing quantitative models can be divided into manual and non-manual. When the stock market first emerged, manual processes were vital. The moving average and PEPB models were manually selected quantitative models and are still used today. The hand-built model adopts the method of "from logic to function". Therefore, the logic and interpretability of each model are solid. The moving average model uses the price moving average. It can filter some noise in the market and generate a clear trend line. The PEPB factor can reflect the company's profitability and the reasonableness of the stock price. However, the artificially established quantitative model has significant limitations, as the stock market is a zero-sum game. If some people make money, some people have to lose money. When a model is widely known, it will inevitably lead some people in the market to use the model to manipulate market sentiment for personal gain. If there are too many of these people, this model will fail. Therefore, obtaining a unique and effective model has become the goal of many quantitative workers. However, the ability of the human brain is ultimately limited. With the development of computer technology, many artificial intelligence algorithms are used to construct quantitative trading models. The algorithms used are usually divided into two categories: neural networks and genetic programming.

2.3.1 Neural Networks (NN)

A Neural Network is a particular type of artificial intelligence. Inspired by the biological nervous system, neural networks combine multiple processing layers through the parallel use of simple element operations. It consists of an input layer, one or more hidden layers and an output layer; each layer uses the previous layer's output as its input[21].

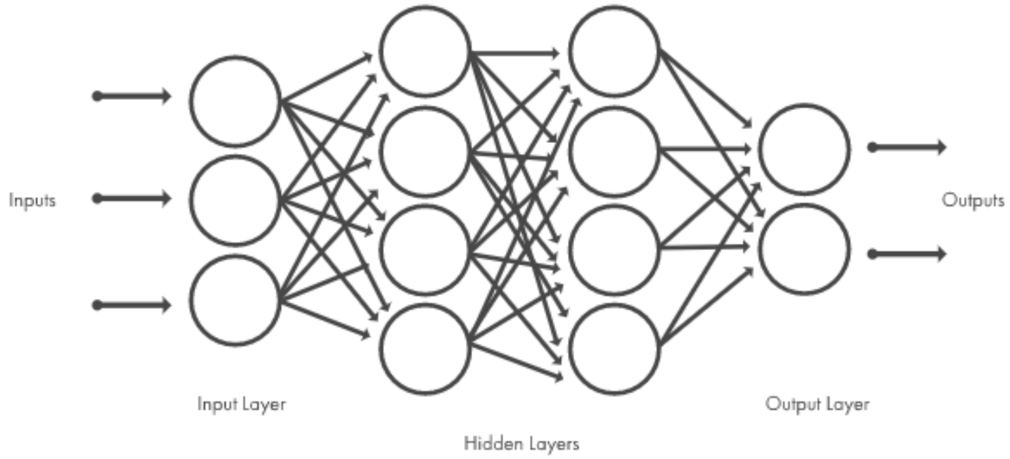


Figure 2.1 Basic Model of NN

Neural Network are often used in algorithmic trading - that is, buying and selling decisions made by algorithmic models. With the help of Neural Networks, computers can realize highly complex functions. Most Neural Network research focuses on predicting the price or trend of stocks or indexes for market timing strategies. According to the structure of the hidden layer, the neural network can be divided into Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Convolutional Neural

Network (CNN) and other types. [21]

LSTM is the most popular Neural Network model among these implementations. LSTM is a special RNN, mainly to solve the problem of gradient disappearance and gradient explosion in the training process of long sequences. Compared with ordinary RNNs, LSTM performs better in longer lines and is very suitable for training stock price prediction. In general, LSTM, which takes a time series consisting of trading indicators based on market microstructure as input, is used to predict stock prices [22]. LSTM can also be used with other neural networks to build a complete trading strategy. For example, in [23], CNN was used for stock selection, LSTM was used for price prediction.

CNN has also been widely used. CNN is characteristically able to extract low-dimensional features from high-dimensional data. It was first used for image recognition. Therefore, researchers tried to use CNN in the field of technical analysis. Technical analysis of stocks is essentially a regression classification of the historical trend of stocks. Due to the excellent characteristics of CNN in image recognition, some studies use neural network models based on CNN. However, to be learned by the CNN model, one-dimensional financial input data needs to be converted into two-dimensional images. Goodluck et al. [24] use technical analysis and clustering to convert the time series of price data into two-dimensional images and use deep CNN for classification to predict stock price trends.

Similarly, Sezer et al. [16] proposed a new technology that converts financial time series data composed of technical analysis indicator output into two-dimensional images and uses CNN to classify these images to determine trading signals. In [24] [25], the researchers used CNN to predict the future trend of stock prices in combination with historical trend of characteristics relating to stock prices. Sezer et al. [26] directly use it to draw images as the input of CNN predict whether the image category is to buy, hold or sell, so the corresponding algorithmic trading model is to be developed.

Deep Neural Networks are often used to build high-frequency trading models. Tran et al. [27] used the DL model to learn the high-frequency limit orders generated in the transaction to predict the stock price trend. In [28], the author uses Fuzzy Deep Direct Reinforcement Learning (FDDR) to predict stock prices and generate open and close position signals.

However, the neural network algorithm has a major issue: it is difficult to find the underlying logic to explain all the situations in the complex stock market. A crucial step in building a neural network algorithm is to adjust the parameters. The return rate of the backtest evaluates the quality of the parameters. But parameters which were the best in the past may not necessarily still be valid in the future. In other words, the model established by the network has a greater risk of failure. Moreover, the failure is difficult to predict, and when the market-style undergoes significant changes, it may cause great losses.

2.3.2 Genetic programming (GP)

Genetic programming (GP) is a well-known intelligent optimization method for seeking optimal solutions, initially proposed by Holland [29] in 1984. Researchers found it very suitable for mining quantitative financial factors in recent years. Many researchers have found that genetic programming can construct factors [30] [31] that are difficult to see through the human brain. Although these factors found by genetic programming are difficult to explain with logic, they can indeed enable users to obtain excess returns on investment. Genetic programming fully utilize a computer’s fast-computing speed, realizing “from Function to Logic”. The factors found in this way also risk failure, but finding new factors is only a matter of computing time. In other words, genetic programming can provide a steady stream of factors for quantitative trading.

In the field of investment, the genetic programming shows extreme effectiveness. According to the survey, some researchers have used them for transactions in various investment products, such as the EUR/USD foreign exchange market [32], the oil futures market [33], and the stock trading markets. In the meantime, it is also applicable to trade in different regions. Some researchers have used it for Chinese stock investment

[33] and American stock investment [34], and both can obtain considerable returns.

The most critical component of genetic programming is the original input data involved in training and the formula that combines various data [29]. Many researchers have pointed out that the use of pre-processed data can significantly improve the reliability of training results [29] [35] [36]. Some researchers have also found that using specific indicators can have unexpected effects, such as the sentiment indicators mentioned by Yang [35] and the RSI, MA, etc., mentioned by Wang [33]. Some investors who use natural language processing to quantify discussions about stocks on the Internet and input them as sentiment indicators, and they have also achieved good results [37] [38].

Some researchers combine genetic programming with other AI algorithms to obtain better returns. Zhang combined genetic programming with Recurrent Reinforcement Learning and found the Sharpe rate of investment can be significantly improved [39]. Kuo combined genetic programming with fuzzy neural networks and artificial neural networks. Since the parameters required by the neural network cannot be determined, the effect was found to be unsatisfactory [40].

The study found that most researchers use genetic programming for daily-level strategies, and almost no researchers use them for intra-day high-frequency systems. One possible reason may be that the intraday data of stocks can better reflect the underlying logic of market transactions, that is, the relationship between supply and demand. If there is an oversupply, prices will fall; in short supply, prices will rise. With this underlying logic, intraday factors should encounter less risk of failure.

2.3.3 Comparison of GP and NN

Both GP and NN are artificial intelligence algorithms suitable for establishing quantitative trading models. The advantage of NN is that there is no need to define the relationship between data. As long as you select a suitable model and use enough data for training, NN will find the proper relationship between the data by itself [41]. However, this is also the shortcoming of NN, because invalid relations may connect the data and thus fall into the local optimum, resulting in an over-fitting situation [42]. In addition, although the Chinese stock market has a history of nearly 30 years, the amount of high-frequency tick-level data is vast. Still, high-frequency information is challenging to be obtained by the public. They are too expensive and monopolized by data providers. Although the daily-level data is cheap, the amount of data is too small to satisfy NN training. Finally, the model trained by NN is a black box, and there is no possibility of using logic to explain it. If the result of the model does not meet expectations, it is difficult to judge whether this is a typical mistake or a failure of the strategy.

Genetic Programming are different. Before GP training, you need to define the relationship between variables and variables. Although this will cause a certain degree of trouble, it saves much time in the training process. The randomness in the GP training process can significantly reduce the possibility of falling into a local optimum. GP training does not need to use too much data. Although the training result is still difficult to interpret compared to the factors selected by the human brain, it is much more interpretable than the black box trained by NN. In the environment of intraday high-frequency data, the transaction logic is infinitely close to the underlying reason. Under this condition, using GP is more likely to obtain a universal quantitative model. Therefore, this article will use GP for building high-frequency quantitative trading factors.

Chapter 3: System Modelling

3.1 Overview

Intraday high-frequency trading, also known as T+0 trading, refers to buying and selling stocks on the same day to earn the difference. However, in the a-share market, T+0 transactions are not allowed. The a-share market adopts the T+1 trading rule. That is, stocks bought today can only be sold tomorrow. Therefore, if high-frequency trading is required in the a-share market, this stock must be held in advance to achieve disguised T+0 trading. It should be noted that stock trading requires commissions, and excessive high-frequency trading will significantly increase commission costs and lead to losses. Therefore, this strategy is more suitable for more volatile stocks during the day.

The model studied here is used to detect when the stock will fluctuate significantly. We call it the big wave factor. Its research process is shown in the following flowchart.

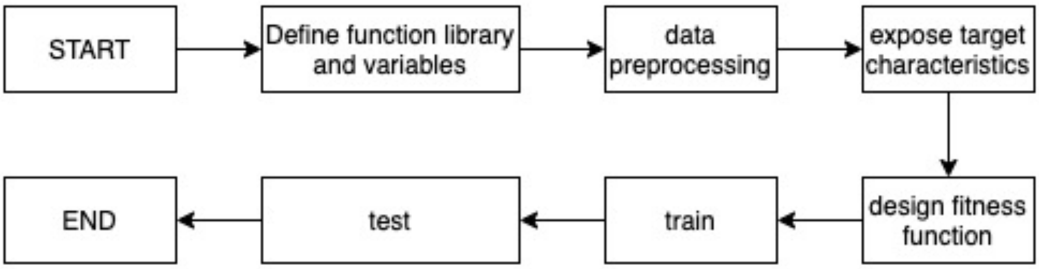

START
Define function library and variables
data preprocessing
expose target characteristics
design fitness function
train
test
END

Figure 3.1 A flowchart of the big wave factor research process

The training data used in this article is the tick data with the stock code 002415 dated

2021/01/25, and the test data is the same stock dated 2021/01/26 and 2021/01/27. The data was obtained from Wind.

3.2 Principles of genetic programming

Mendel published a paper entitled “Experiments in plant hybridisation” in 1866. In this paper, he put forward essential concepts such as genes and clarified the laws of heredity. He believed that the genes of two parents would combine when they reproduce the next generation, thereby producing offspring that are not precisely the same as the parents, forming a diversity of biological individuals. Darwin’s “On the Origin of Species” argues that the traits of individuals who can better adapt to the environment are retained through the natural selection of diverse individuals. Individuals who cannot adapt to the climate face elimination. The genes in the population that can adapt to the environment are included, and the entire population can be better adapted to the climate [43].

The genetic programming simulates biological genetics, hybridisation, and natural selection processes. In genetic programming, each solution is called an individual. It represents a variable sequence; we can think of it as a gene or chromosome. To facilitate the evolution of formulas, individuals in genetic programming are generally represented in the form of a binary tree. Suppose there is an individual G whose chromosome contains two variables, X and Y. A possible expression is: G = (X + 3) *

(Y – 2). In the genetic programming, the above expression is represented by S-expression: G = (*(+X3) (-Y2)); we can express the formula as a binary tree, as shown in Figure 3.2:

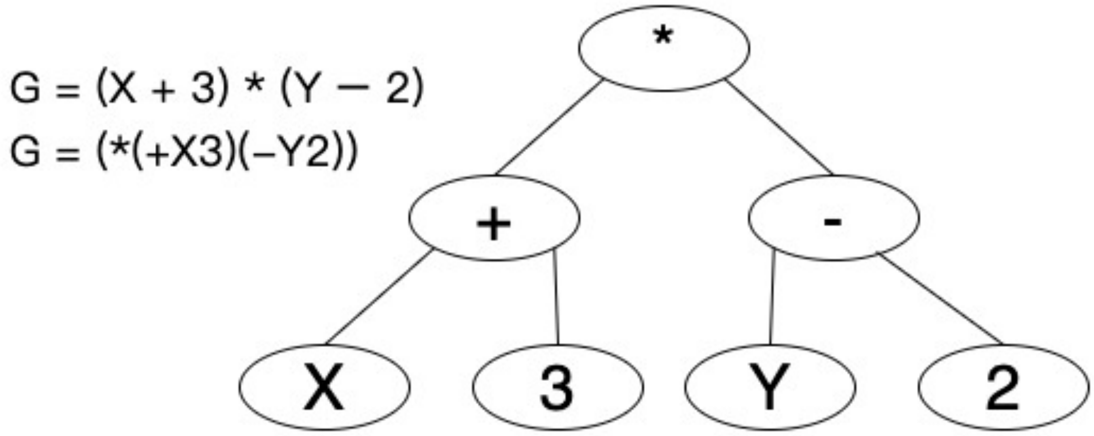


Figure 3.2 S-expression and binary tree

All the leaves are variables or constants in this binary tree, and the internal nodes are functions. Any subtree in the tree can be modified or replaced. The output value of the formula can be obtained by the recursive method.

First, the algorithm will randomly generate a certain number of individuals as the initial value of the population, use the fitness function to evaluate each individual's fitness and select them. The selection is based on the individual's fitness, but it does not mean that it is entirely oriented to the level of fitness because simply selecting the individual with the highest fitness may cause the algorithm to quickly converge to the optimal local solution instead of the optimal global solution. Therefore, the method of genetic programming selection is that the higher the fitness, the higher the probability of being selected; the lower the fitness, the lower the likelihood of being selected.

The next step is to make the selected individuals produce the next generation and form a new population. This process is called reproduction. The reproduction process includes operations such as crossover and mutation. Crossover refers to allowing two selected individuals to exchange gene segments. The genetic programming in this paper sets a crossover probability, which reflects the likelihood that two selected individuals will mate. For example, the mating probability is 0.9, and every two individuals have a 90% probability of mating to produce two new individuals, replacing the original "old" individuals. The crossover process will generate a crossover point at any position in each chromosome. The parental chromosomes break at the crossover point and exchange fragments. As shown in Figure 3.3:

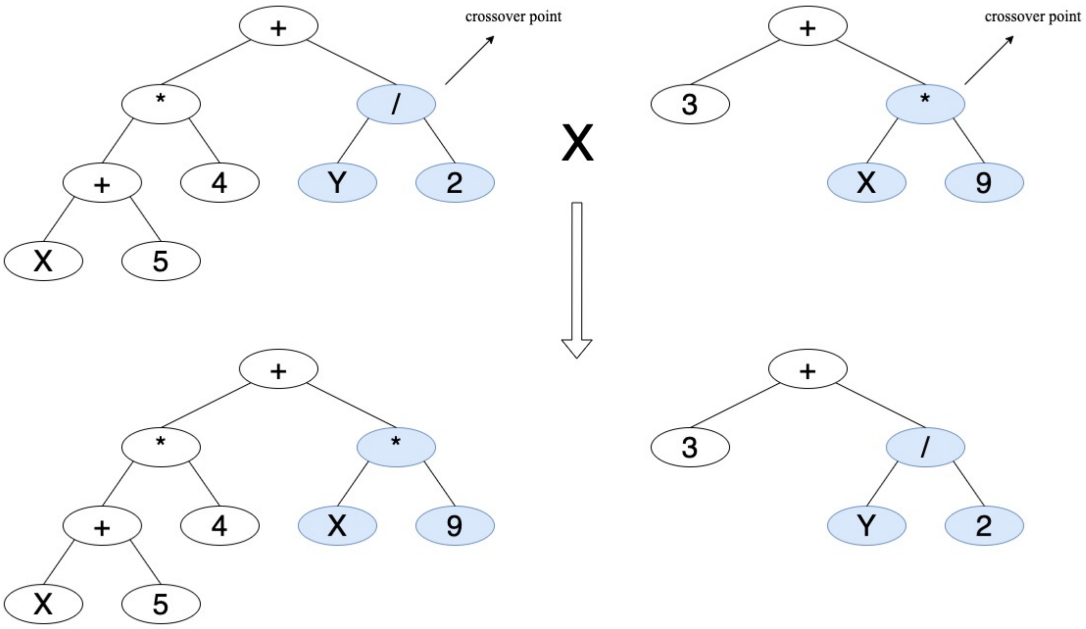


Figure 3.3 Crossover

Mutation refers to changing a particular gene segment after creating a new individual. In this project, a mutation probability is designed for conversion. According to this probability, the chromosome of the unique individual undergoes random transformations, introducing new genes into the entire population. As shown in Figure 3.4:

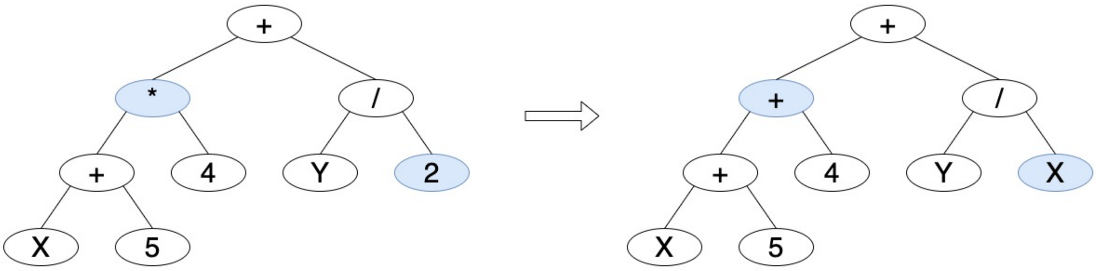


Figure 3.4 Mutation

After a series of selection processes, mating and mutation, the population will develop in the direction of higher overall fitness from generation to generation. This process is repeated continuously: evaluating each individual, calculating fitness, mating in pairs, and then mutating to produce the next generation until the termination conditions are met. The general termination conditions are as follow [45]:

- Computing resource constraints (e.g., computing time, memory occupied by computing, etc.)
- An individual has met the condition of the optimal value. That is, the optimal weight has been found.
- Fitness has reached saturation and continuing to evolve will not produce individuals with better fitness.
- Human intervention

Fitness saturation is selected as the termination condition. This termination condition has a higher probability of obtaining the optimal global solution, but the training lasts longer. The flowchart of the entire genetic programming is shown in Figure 3.5.

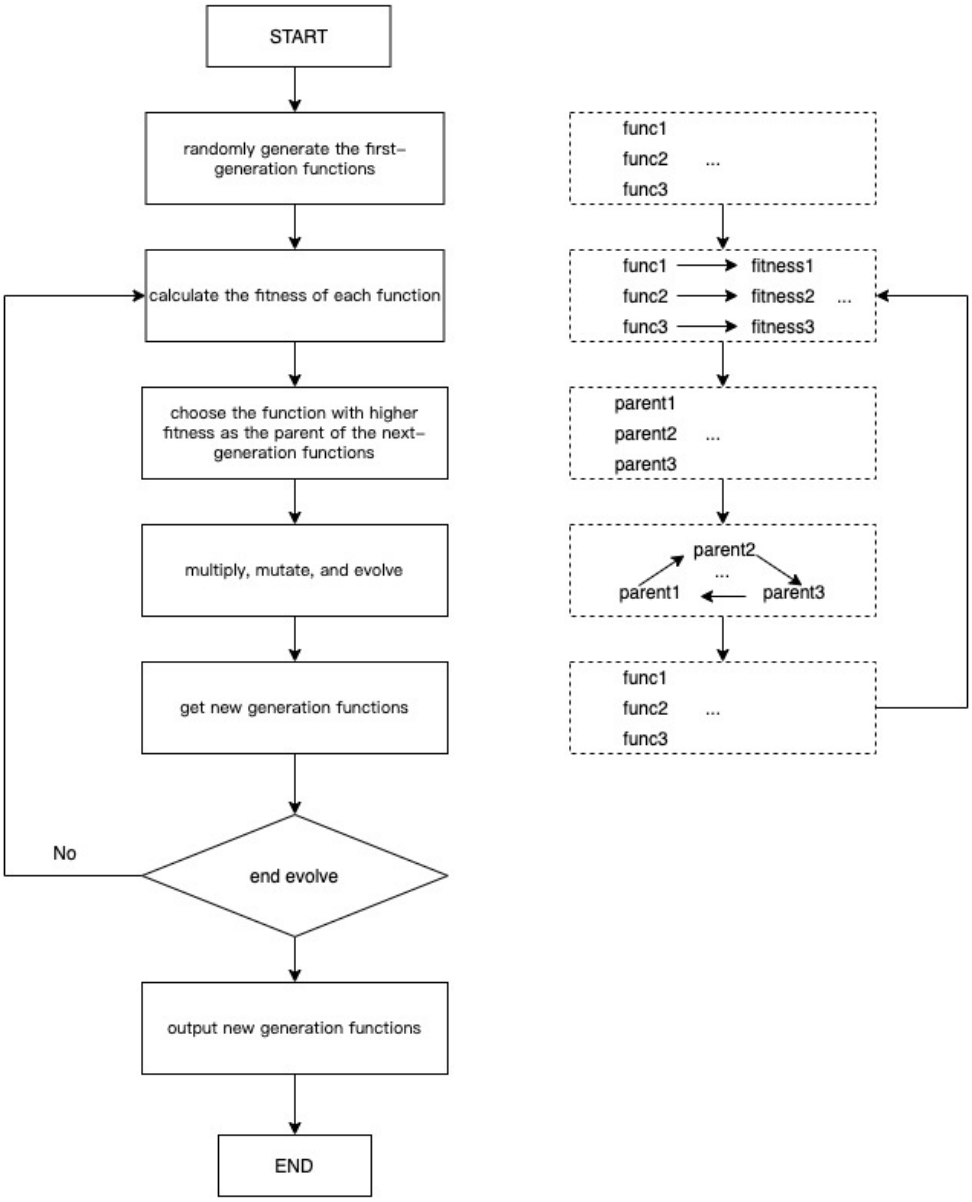

START
randomly generate the first-generation functions
calculate the fitness of each function
choose the function with higher fitness as the parent of the next-generation functions
multiply, mutate, and evolve
get new generation functions
end evolve
No
output new generation functions
END
func1
func2 ...
func3
func1 → fitness1
func2 → fitness2 ...
func3 → fitness3
parent1
parent2 ...
parent3
parent2
...
parent1
parent3
func1
func2 ...
func3

Figure 3.5 The flow of the genetic programming

3.3 Statistical summary

The stock studied in this article is Hikvision (002415.SZ). The following is a statistical analysis of the closing price of the stock from January 2, 2018 to January 27, 2021.

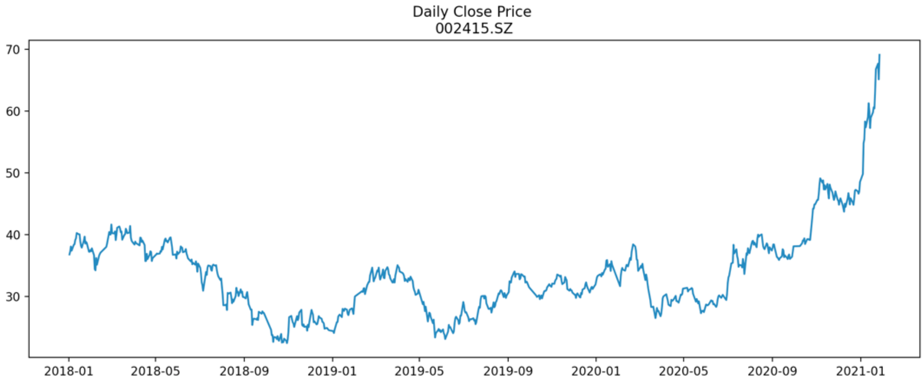


Figure 3.6 Daily close price

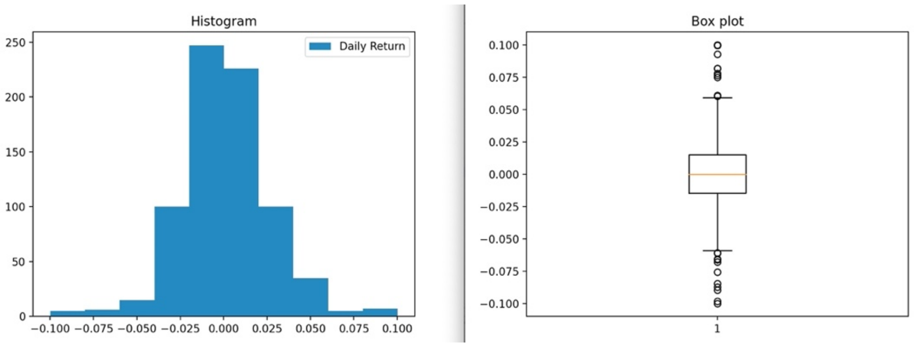


Figure 3.7 Daily return histogram and boxplot

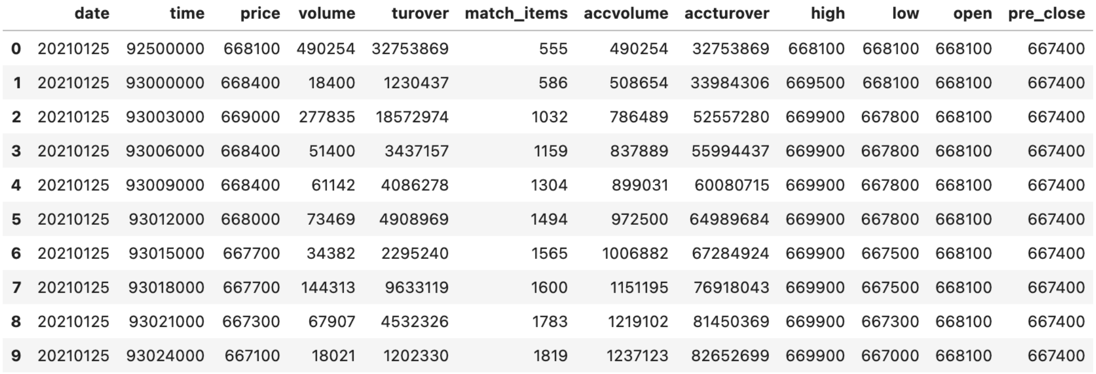
Mean	Median	Standard deviation
0.0011829	0.0001115	0.0258884

Table 3.1 Statistics of daily return

3.4 Data overview

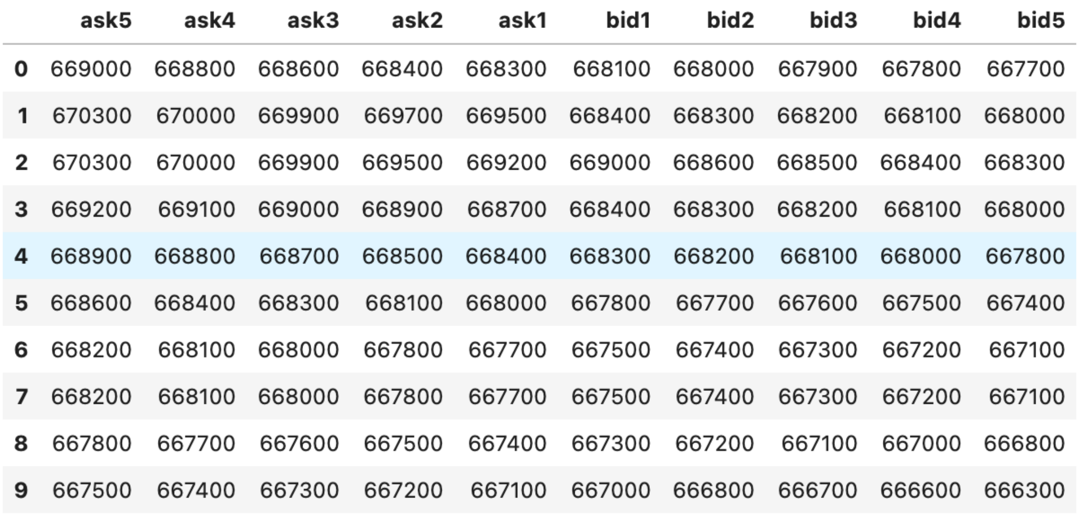
After removing some deprecated fields and having no apparent value for this study from the high-frequency tick data obtained by wind, the remaining table has 4763 rows and 36 columns. The tick data is a 3-second snapshot of the market, and each row represents the transactions that occurred within 3 seconds.

The figure below illustrates some of the natural high-frequency data.



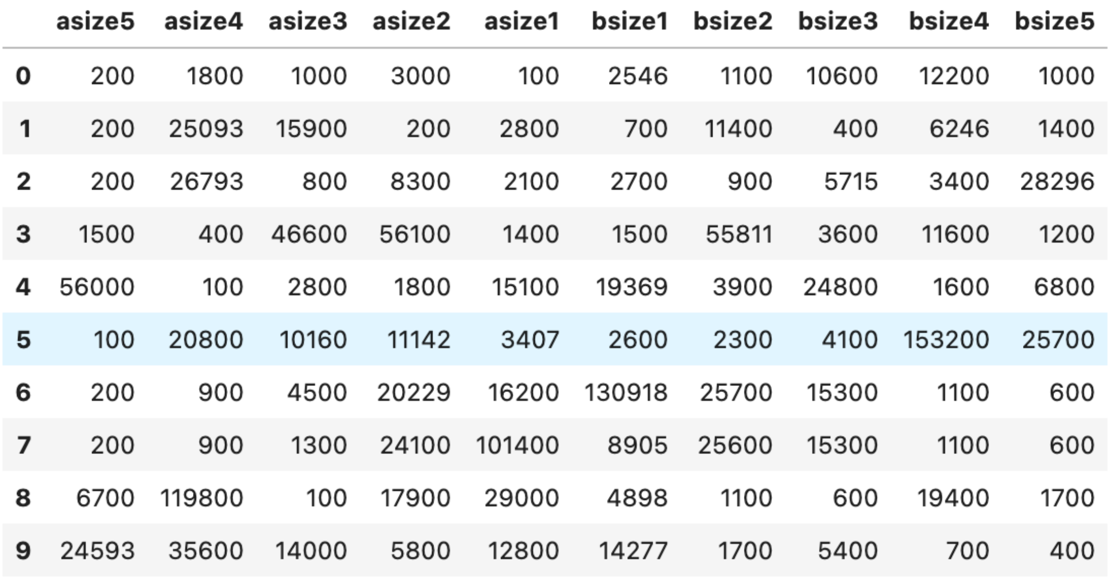
	date	time	price	volume	turover	match_items	accvolume	accturover	high	low	open	pre_close
0	20210125	92500000	668100	490254	32753869	555	490254	32753869	668100	668100	668100	667400
1	20210125	93000000	668400	18400	1230437	586	508654	33984306	669500	668100	668100	667400
2	20210125	93003000	669000	277835	18572974	1032	786489	52557280	669900	667800	668100	667400
3	20210125	93006000	668400	51400	3437157	1159	837889	55994437	669900	667800	668100	667400
4	20210125	93009000	668400	61142	4086278	1304	899031	60080715	669900	667800	668100	667400
5	20210125	93012000	668000	73469	4908969	1494	972500	64989684	669900	667800	668100	667400
6	20210125	93015000	667700	34382	2295240	1565	1006882	67284924	669900	667500	668100	667400
7	20210125	93018000	667700	144313	9633119	1600	1151195	76918043	669900	667500	668100	667400
8	20210125	93021000	667300	67907	4532326	1783	1219102	81450369	669900	667300	668100	667400
9	20210125	93024000	667100	18021	1202330	1819	1237123	82652699	669900	667000	668100	667400

Figure 3.8 Tick Data from column 1 to column 12



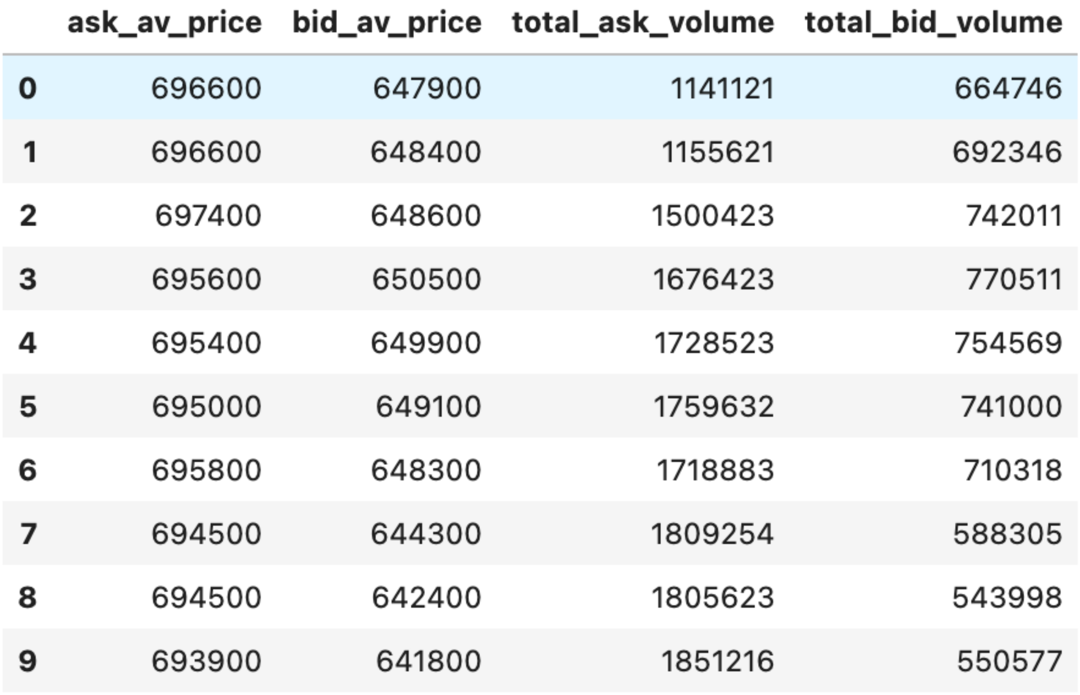
	ask5	ask4	ask3	ask2	ask1	bid1	bid2	bid3	bid4	bid5
0	669000	668800	668600	668400	668300	668100	668000	667900	667800	667700
1	670300	670000	669900	669700	669500	668400	668300	668200	668100	668000
2	670300	670000	669900	669500	669200	669000	668600	668500	668400	668300
3	669200	669100	669000	668900	668700	668400	668300	668200	668100	668000
4	668900	668800	668700	668500	668400	668300	668200	668100	668000	667800
5	668600	668400	668300	668100	668000	667800	667700	667600	667500	667400
6	668200	668100	668000	667800	667700	667500	667400	667300	667200	667100
7	668200	668100	668000	667800	667700	667500	667400	667300	667200	667100
8	667800	667700	667600	667500	667400	667300	667200	667100	667000	666800
9	667500	667400	667300	667200	667100	667000	666800	666700	666600	666300

Figure 3.9 Tick Data from columns 13 to column 22



	asize5	asize4	asize3	asize2	asize1	bsize1	bsize2	bsize3	bsize4	bsize5
0	200	1800	1000	3000	100	2546	1100	10600	12200	1000
1	200	25093	15900	200	2800	700	11400	400	6246	1400
2	200	26793	800	8300	2100	2700	900	5715	3400	28296
3	1500	400	46600	56100	1400	1500	55811	3600	11600	1200
4	56000	100	2800	1800	15100	19369	3900	24800	1600	6800
5	100	20800	10160	11142	3407	2600	2300	4100	153200	25700
6	200	900	4500	20229	16200	130918	25700	15300	1100	600
7	200	900	1300	24100	101400	8905	25600	15300	1100	600
8	6700	119800	100	17900	29000	4898	1100	600	19400	1700
9	24593	35600	14000	5800	12800	14277	1700	5400	700	400

Figure 3.10 Tick Data from column 23 to column 32



	ask_av_price	bid_av_price	total_ask_volume	total_bid_volume
0	696600	647900	1141121	664746
1	696600	648400	1155621	692346
2	697400	648600	1500423	742011
3	695600	650500	1676423	770511
4	695400	649900	1728523	754569
5	695000	649100	1759632	741000
6	695800	648300	1718883	710318
7	694500	644300	1809254	588305
8	694500	642400	1805623	543998
9	693900	641800	1851216	550577

Figure 3.11 Tick Data from column 33 to column 36

All data can be divided into three categories: time, price, and volume from the dimension of attributes. The data of the time attribute only represents the time when the transaction occurred and is not within the calculation scope of factor mining, so it will not participate in the operation of the genetic programming.

The following table explains the fields of the data.

variables name	**definition**
price	Latest price
volume	Latest volume

turnover	Transaction amount
match_items	Number of transactions
accvolume	Total volume of the day
accturnover	Total turnover of the day
high	Highest price of the day
low	Lowest price of the day
open	Open price of the day
pre_close	Closing price of yesterday
ask1 ~ ask5	Ask 1 price to Ask 5 price
bid1 ~ bid5	Bid 1 price to Bid 5 price
asize1 ~ asize5	Ask 1 volume to Ask 5 volume
bsize1 ~ bsize5	Bid 1 volume to Bid 5 volume
ask_av_price	Weighted average order sell price
bid_av_price	Weighted average order buy price
total_ask_volume	Amount of order sell
total_bid_volume	Amount of order buy

Table 3.2 Description of variables

3.4 Defining function library

Functions and variables are the components of individuals. As mentioned above, all the leaves of each individual are variables or constants, and the internal nodes are functions.

In the following description, capital letters such as "X" and "Y" are vectors which represent the factor value of the current stock in the specified time window. Lowercase letters such as "d" are float type data.

functions name	**definition**
add (X, Y)	Return a vector, the i-th element is X_i + Y_i
sub (X, Y)	Return a vector, the i-th element is X_i - Y_i
mul (X, Y)	Return a vector, the i-th element is X_i * Y_i
div (X, Y)	Return a vector, the i-th element is X_i / Y_i
sqrt (X)	Return a vector, the i-th element is sqrt (X_i)
log (X)	Return a vector, the i-th element is log (X_i)
abs (X)	Return a vector, the i-th element is | X_i |

neg (X)	Return a vector, the i-th element is -X_i
inv (X)	Return a vector, the i-th element is X_{n-i}, n = len (X)
max (X, Y)	Return a vector, the i-th element is the larger one in X_i and Y_I
min (X, Y)	Return a vector, the i-th element is the smaller one in X_i and Y_I
v_max (X)	Return an integer, the value is the largest one in X
v_min (X)	Return an integer, the value is the smallest one in X
pct_change (X)	Return a vector, the i-th element is (X_i - X_{i-1}) / X_{i-1}
w_sum (X, n)	Return a vector, the i-th element is $\sum_{k=0}^{n-1} X_{i-k}$
w_sma (X, n)	Return a vector, the i-th element is $\frac{\sum_{k=0}^{n-1} X_{i-k}}{n}$
stddev (X, n)	Return a vector, the i-th element is $\sqrt{\frac{\sum_{k=0}^{n-1} (X_{i-k}-\overline{X})^2}{n}}$, $\overline{X} = \frac{\sum_{k=0}^{n-1} X_{i-k}}{n}$
arg_max (X)	Return a vector, the i-th element is the index of largest value in X_i

arg_min (X)	Return a vector, the i-th element is the index of smallest value in X_i

Table 3.3 Description of functions

3.5 Preprocessing data

Since “Tick” contains different types of data, such as price, volume, etc., direct participation in the calculation will incur problems such as inconsistent units, so they need to be normalized. The normalization method in this article uses Min-Max scaling, and the formula is as follows.

$$X_{norm\ i} = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$

Function 3.1 Normalization function

This method implements equal scaling of the original data, where $X_{norm\ i}$ is the normalized data, X_i is the actual data, and X_{max} and X_{min} are the maximum and minimum values of the X.

It is worth noting that future data should not be introduced here. For example, the highest and lowest prices of the day cannot be known in the intraday market. So the data used when normalizing here are the maximum and minimum values of the previous trading day; besides, in this article, all data related to real-time prices, such as open, high, low, ask1 ~ 5, etc., will be classified into one category and normalized together. Similarly, data related to real-time trading volume will also be organized for normalization.

3.6 Exposing target characteristics

The set target of a general genetic programming is the stock price or the change in the stock price over a certain period of time. Still, experiments found that the result obtained by taking the stock price as the target is not ideal, and it is difficult to employ as a basis for trading.

Because too many factors affect stock prices, stock prices can be regarded as a cross-section of multi-dimensional data. In the case of insufficient data, blindly increasing variables will only increase the complexity of training, and it will be difficult to obtain reliable results. Many researchers [44] compare the linear correlation coefficient between the price curve and the factor value curve. The linear correlation coefficient is between -1 and 1. When the linear correlation coefficient equals 1, the two curves are completely linearly correlated. They find the factor with a more considerable absolute value of the linear correlation coefficient with the stock price curve through training. This method reduces the multi-dimensional price to a two-dimensional linear correlation.

The big wave factor studied in this article aims to identify the time point where the stock price will change drastically. It further reduces the training target to a one-dimensional time point, which improves the success rate of training. To expose the character of huge fluctuation, this article adopts the method of detecting the range of price (Max price – Min price) within 500 Tick, as shown in Figure 3.10, drawing the range line with the window of 500.

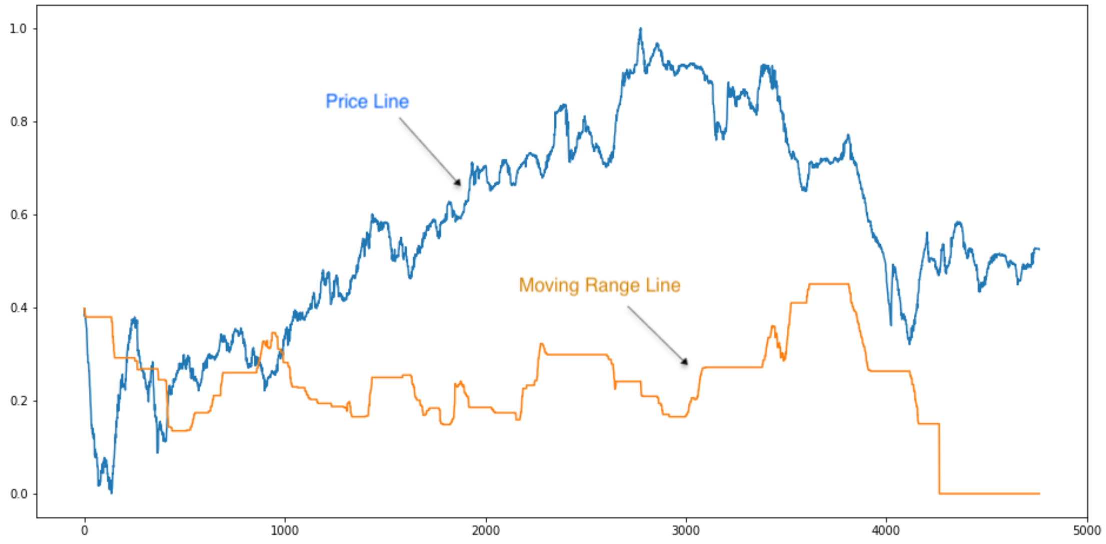


Figure 3. 12 Moving Range Line

The reason for the “platforms” is that there is no new, more extensive range in this window, so the endpoint of each “platform” means the end of a band so that they will be regarded as the target points. To eliminate tiny fluctuations, this project selects only the top 30% range of the band, and the chosen target points are shown in figure 3.11.

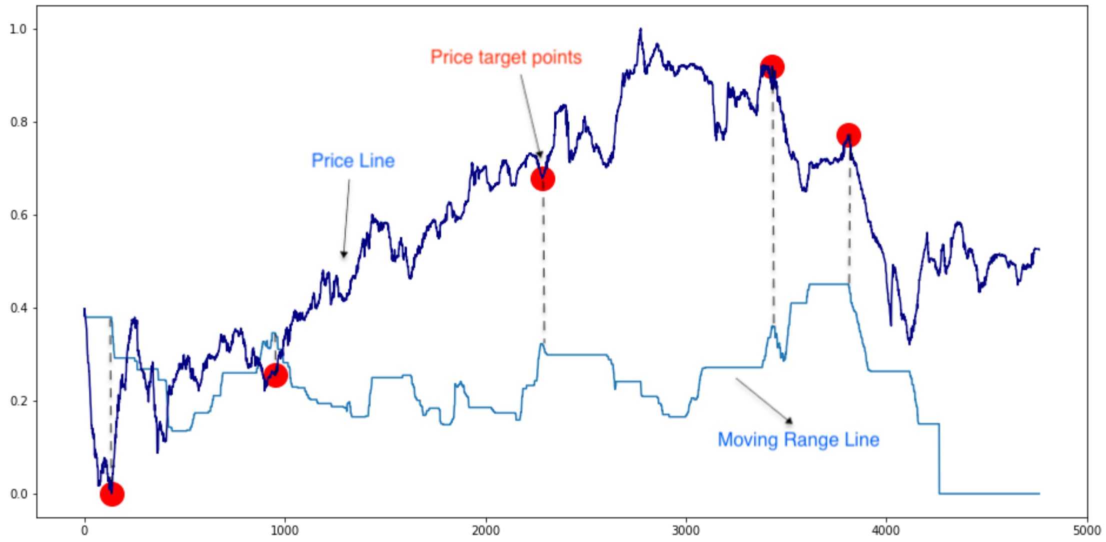


Figure 3.13 Target points of Moving Range Line

3.7 Designing fitness function

The fitness function is the key to a genetic programming. The fitness function is as follows: first, we will use the same method as the previous section to draw the range of the factor function moving 500 ticks and use the same way to filter out the factor target points and match the nearest price target point for each factor target point. The fitness is the average of the distances between all matched target points; the lower the fitness, the higher the adaptability. As shown in the following function.

$$fitness = \frac{\sum_{x=1}^{n} Min(abs(Tf_x * E - Tp))}{n}$$

Function 3.2 Fitness function

Where Tf is the vector of the set of factor target points' abscissa, Tp is the vector of the group of price target points' abscissa, E is a vector with the same dimension as Tp, and all elements are 1.

The reason for designing the average value is that the number of target points selected by the same method may not have the exact count. If the average value is not used, the adaptability of the factor that selects only a few points will be much greater than that of the majority of points, which is unacceptable.

3.8 Parameter settings

The main parameters of the genetic programming in this paper are set as shown in the table below. The parameters in the table and their definitions are from the official documentation of gplearn. [46]

Parameter name	**Definition**	**Parameter settings**
population_size	The number of programs in each generation.	10000
generations	The number of generations to evolve.	20
tournament_size	The number of programs that will compete to become part of the next generation.	20
stopping_criteria	The required metric value required in order to stop evolution early.	0.0
init_depth	The range of tree depths for the initial population of naive formulas. Individual trees will randomly choose a maximum depth from this range.	(3, 6)
p_crossover	The probability of performing crossover on a tournament winner. Crossover takes the winner of a tournament and selects a random	0.9

	subtree from it to be replaced. A second tournament is performed to find a donor. The donor also has a subtree selected at random and this is inserted into the original parent to form an offspring in the next generation.	
p_subtree_mutation	The probability of performing subtree mutation on a tournament winner. Subtree mutation takes the winner of a tournament and selects a random subtree from it to be replaced. A donor subtree is generated at random and this is inserted into the original parent to form an offspring in the next generation.	0.01
p_hoist_mutation	The probability of performing hoist mutation on a tournament winner. Hoist mutation takes the winner of a tournament and selects a random subtree from it. A random subtree of that subtree is then selected and this is 'hoisted' into the original subtrees location to form an offspring in the next generation. This method helps to control bloat.	0.01

p_point_mutation	The probability of performing point mutation on a tournament winner. Point mutation takes the winner of a tournament and selects random nodes from it to be replaced. Terminals are replaced by other terminals and functions are replaced by other functions that require the same number of arguments as the original node. The resulting tree forms an offspring in the next generation.	0.01
p_point_replace	For point mutation only, the probability that any given node will be mutated.	0.05

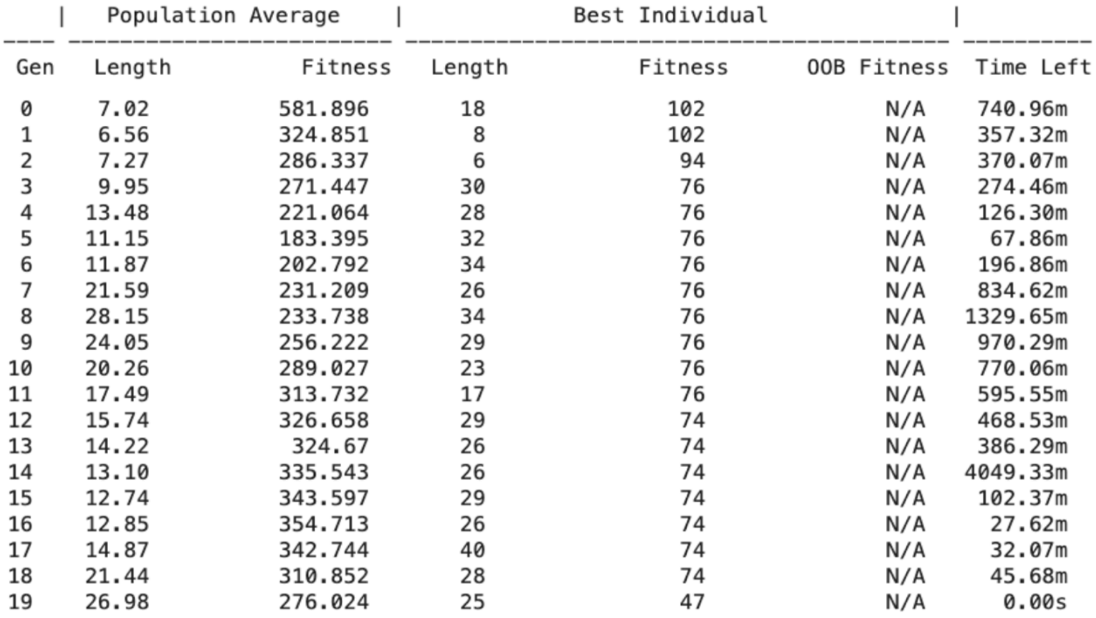
Table 3.4 Parameter Definition and Settings

Other parameters not declared in the paper used the default values of gplearn.

Chapter 4: Model training and testing

4.1 Model training

The evolution process is shown in figure 4.1.



```
    |   Population Average    |             Best Individual              |
---- ------------------------- ------------------------------------------ ----------
 Gen   Length          Fitness   Length          Fitness      OOB Fitness  Time Left
   0     7.02          581.896       18              102              N/A    740.96m
   1     6.56          324.851        8              102              N/A    357.32m
   2     7.27          286.337        6               94              N/A    370.07m
   3     9.95          271.447       30               76              N/A    274.46m
   4    13.48          221.064       28               76              N/A    126.30m
   5    11.15          183.395       32               76              N/A     67.86m
   6    11.87          202.792       34               76              N/A    196.86m
   7    21.59          231.209       26               76              N/A    834.62m
   8    28.15          233.738       34               76              N/A   1329.65m
   9    24.05          256.222       29               76              N/A    970.29m
  10    20.26          289.027       23               76              N/A    770.06m
  11    17.49          313.732       17               76              N/A    595.55m
  12    15.74          326.658       29               74              N/A    468.53m
  13    14.22           324.67       26               74              N/A    386.29m
  14    13.10          335.543       26               74              N/A   4049.33m
  15    12.74          343.597       29               74              N/A    102.37m
  16    12.85          354.713       26               74              N/A     27.62m
  17    14.87          342.744       40               74              N/A     32.07m
  18    21.44          310.852       28               74              N/A     45.68m
  19    26.98          276.024       25               47              N/A      0.00s
```

Figure 4.1 Evolution process

It can be seen from the evolution process that as the factors continue to iterate, the fitness of the factors gets higher. However, as mentioned above, the genetic programming is essentially a search algorithm, and the time complexity is relatively high. This training took 67 hours using Intel Core i7 Processor (2.5GHz). If using a faster CPU, the training time may be shortened.

4.2 Results

The result of genetic programming training can be regarded as a formula, and the factor value can be output as long as the required data is input. Since the factor obtained in this article has been adopted by a private equity institution and was signed by a confidentiality agreement, the specific formula content will not be disclosed. However, this paper will show the factor values in the train and test set for research purposes.

The factor values in the training set (date: 2021/01/25) are shown in the following figure.

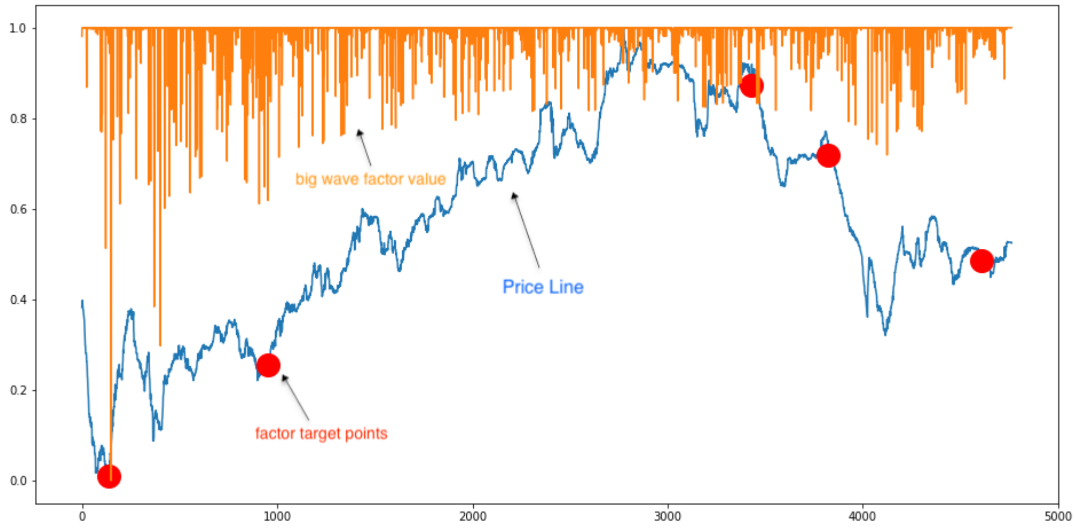


Figure 4.2 Big wave factor value and factor target points

(stock code: 002415, date: 2021/01/25)

The factor values in the test set (date: 2021/01/26 and 2021/01/27) are shown in the following figure.

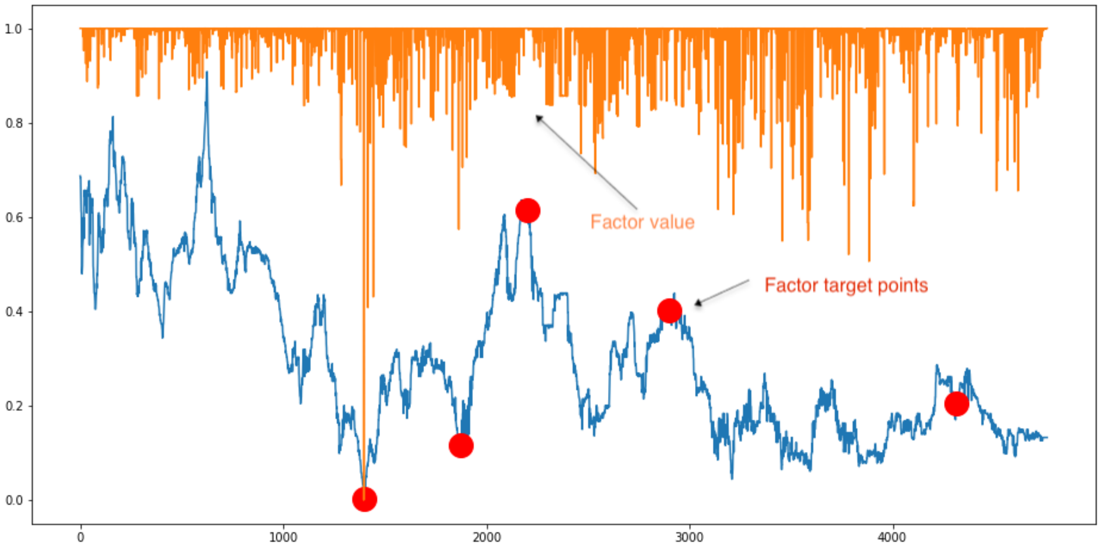


Figure 4.3 Big wave factor value and factor target points

(stock code: 002415, date: 2021/01/26)

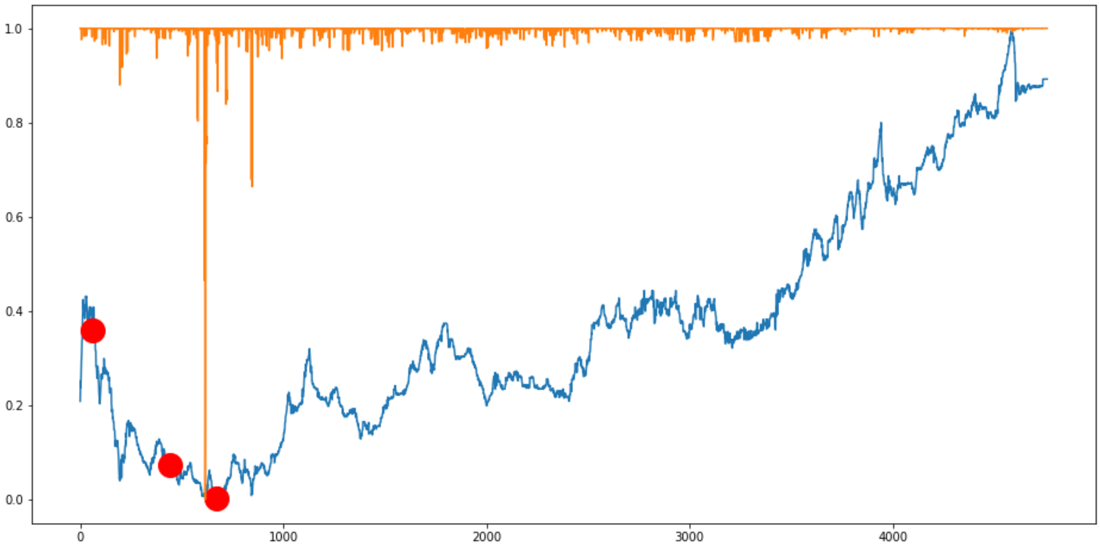


Figure 4.4 Big wave factor value and factor target points

(stock code: 002415, date: 2021/01/27)

It can be seen from the results that this factor basically screens out the target points. Although some issues are missed, such a signal can obtain benefits.

4.3 Strategy Design and Benefit Estimation

From the analysis of the signal characteristics of the big wave factor, its advantage is that it can indicate the time point when there may be significant fluctuations in the future. Its disadvantage is that it cannot judge the direction of changes nor the persistence of fluctuations.

In the stock market, price fluctuations have a momentum effect [11]. The specific performance is that the stock's rate of return tends to continue the original movement direction. This solves the core problem that the big wave factor cannot judge the direction. Set a threshold **m** for momentum strategy. If the stock price change rate exceeds **m** in time **t**, it is considered that the stock price will change in this direction.

Another problem is when to close a position. Since the factor returns studied in this paper come from the short-term trend of the stock price, the timing of closing the position should be the period when the short-term trend ends. Therefore, this paper sets the retracement parameter **r**, and when the stock price retraces **r** from the highest point of profit after opening the position, the position should be closed.

The parameters settings in the test are as follows.

Parameter names	parameters settings
m	0.2%
t	3
r	0.2%

Table 4.1 The parameters and parameter settings in strategy

The strategy designed in this paper using the superimposed momentum effect of the big wave factor is as follows: after the big wave factor prompts the signal, wait for the stock price to fluctuate by 0.2%. If the stock price rises by 0.2%, go long; If the stock price falls by 0.2%, go short. If the stock price does not fluctuate by 0.2% in any direction within 3 minutes, cancel the transaction. Close the position when the retracement exceeds 0.2%.

This strategy can obtain greater profits when the target has major fluctuations, suitable for the big wave factor. However, the shortcomings of this strategy are also evident. The momentum strategy based on the percentage of price change is prone to frequent signals when the stock price is too low. For example, the stock price of the Agricultural Bank of China (601288) is about 3 RMB, and the slightest change in the a-share market price is 0.01 RMB, which means that the momentum strategy will send a signal every time the price changes. In that case, the momentum strategy is ineffective. Therefore, the range of stock prices should be limited before implementing this strategy.

Backtesting using the strategies mentioned above, after estimation, the big wave factor gains 7.4% on 25 January 2021. This seems somewhat high, but it should be noted that the data used for training is the data of the a-shares, which does not support the T+0 strategy. This means that if you want to open a short position, you must open a position in advance, and the position used for opening a position will greatly affect the final return. Therefore, this article has devised a better method for evaluating intraday returns. The premise of this strategy is that all positions remain unchanged before the market closes. Assuming that the opening is 1,000 shares, there must be 1,000 shares at the close. Under this premise, the income can be estimated by calculating the change in the cost of holding positions. The cost calculate function is as follows.

$$Cost = \frac{total\ buy\ amount - total\ sell\ amount + \ stamp + handling}{position}$$

Function 4.1 Cost calculate function.

Using this estimation method, assuming completed buying 002415 at the closing time of the date 2021/01/22, the position is opened for 10,000 shares, the price is 66.17, the handling fee is three ten thousandths, and the stamp duty is one-thousandth. The total purchase amount is 661700. The holding cost is 66.19, and only 1,000 shares are traded in each intraday transaction. Under such a setting, ultimately relying on this strategy to reduce the cost from 66.19 to 65.96, the actual rate of return is 0.34%. Although this seems quite low, this is the essence of high-frequency trading with quantitative strategies: accumulating less into more.

On 2021/01/26, assuming that the position on the renewal date of 2021/01/25 continues to trade, the strategy reduces the cost from 65.96 to 65.72, and the rate of return is 0.36%. On the date of 2021/01/27, this strategy reduces the cost from 65.72 to 65.6, and the rate of return is 0.3%. Surprisingly, although this factor sometimes fails when detecting large fluctuations, after several tests, it is found that the factor value becomes extremely small when the stock price is at the lowest point of the day, which gives us sufficient confidence in the factor.

4.4 Comparison with MACD

Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator often used in trend trading strategies. This paper designed a MACD strategy for comparison. Draw the intraday 10-minute and 30-minute moving averages. When the 10-minute line crosses the 30-minute line upwards, go long, and when the 10-minute line crosses the 30-minute line downwards, close the position. The position must remain unchanged before the market closes, as shown in Figure 4.5.

Date 2021/01/25

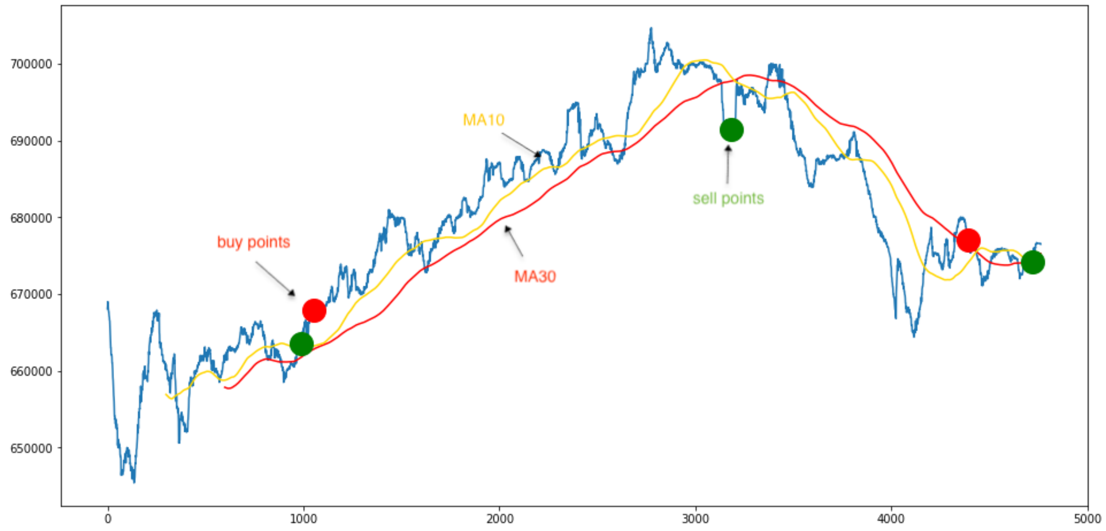


Figure 4.5 MACD strategy schematic diagram 002415-2021/01/25

Using the same parameters as the big wave factor, the MACD strategy reduces the cost from 66.19 to 66.00. The actual rate of return is 0.29%, slightly lower than the big wave factor strategy.

Date 2021/01/26

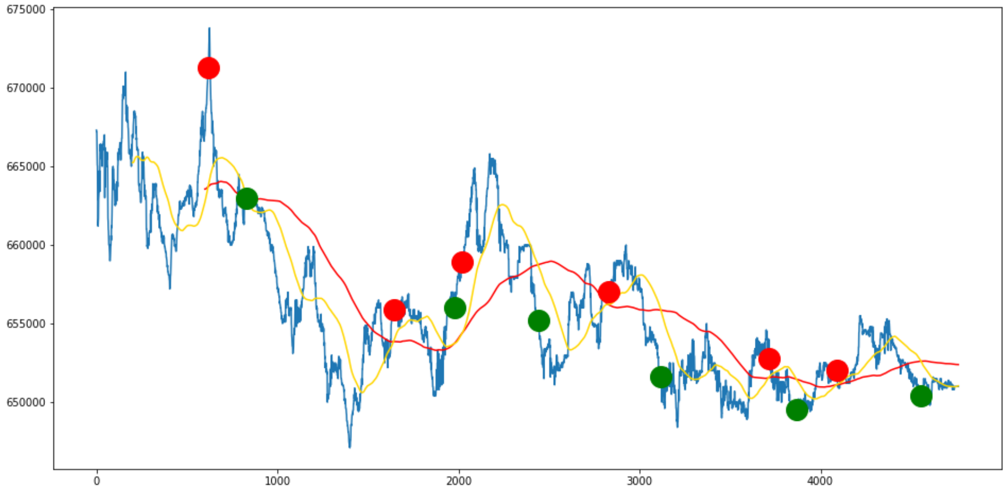


Figure 4.6 MACD strategy schematic diagram 002415-2021/01/26

As shown in this figure, due to the rapid fluctuation of stock prices that day, the shortcomings of the delay of the moving average system are fully exposed. The MACD strategy performed very severely during the day, and in fact, no operation was profitable. The strategy increased the cost from 66.00 to 66.28, and the rate of return was -0.42%.

Date 2021/01/27

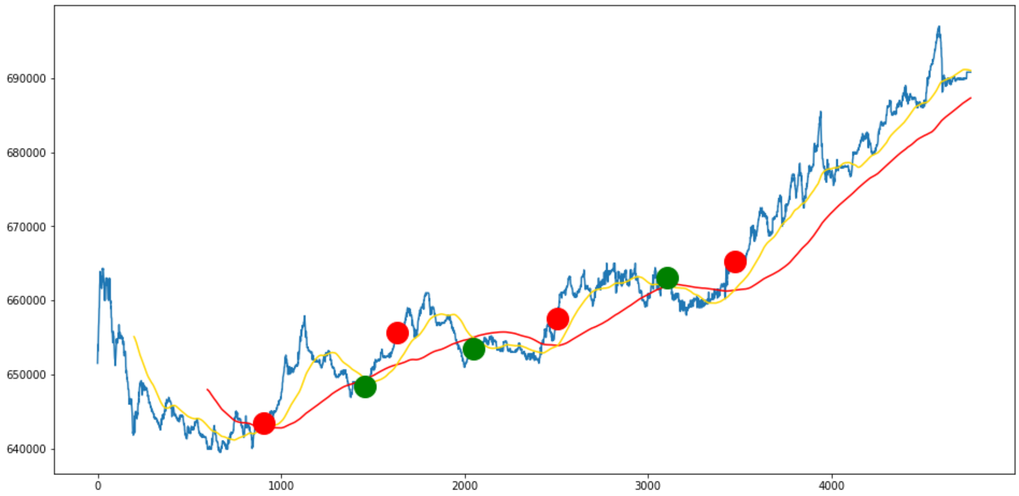


Figure 4.7 MACD strategy schematic diagram 002415-2021/01/27

Since there was a major rising wave in the afternoon that day, the advantages of the moving average system were also exposed. It can continue to make profits during a significant trend. The strategy reduces the cost from 66.28 to 65.99, and the rate of return is 0.44%.

As shown in the cost curve in the figure below, the big wave factor strategy has made stable profits in three days without being disturbed by intra-day fluctuations, and there is almost no delay. It is suitable for a variety of market conditions. Compared to the traditional moving average system strategy, the big wave factor strategy has obvious advantages.

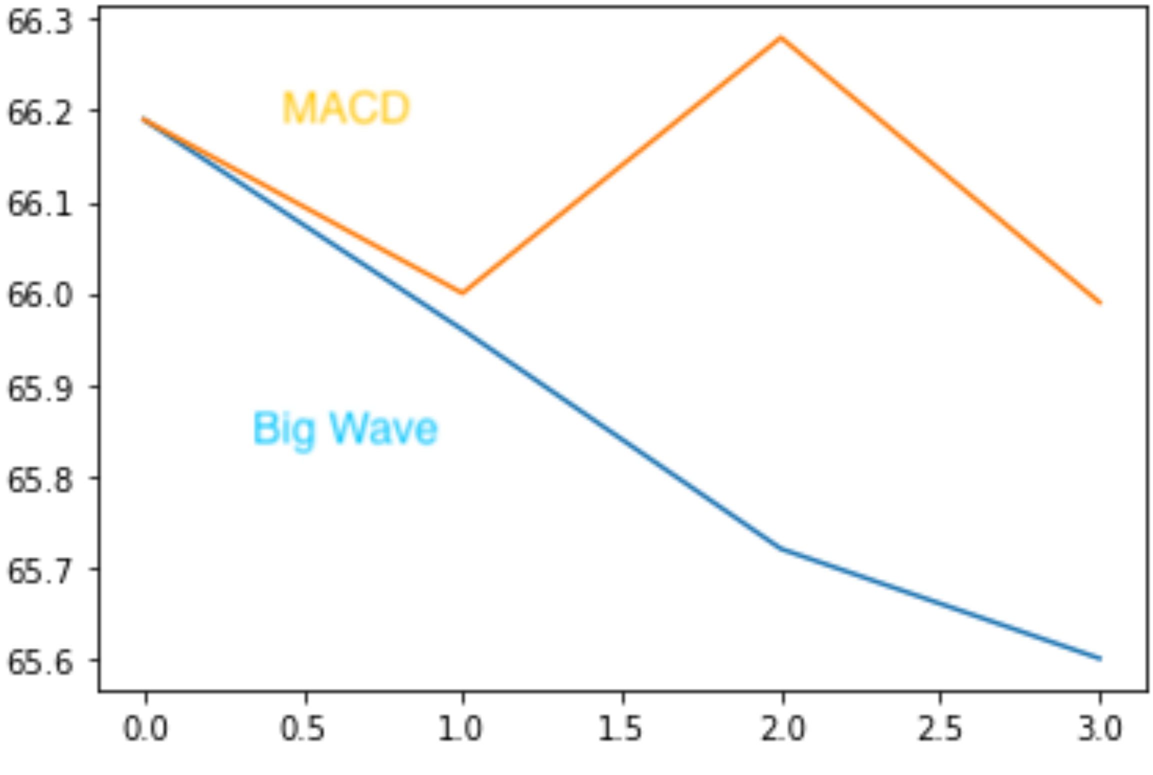


Figure 4.8 Cost curve of Big Wave factor strategy and MACD strategy

As a traditional trend-following trading indicator, MACD can be very profitable when the direction of market volatility is clear. However, any trend-following strategy has a drawback: the trading is lagging. This lag can lead to huge losses when market volatility is small and frequent. The trading strategy proposed in this paper uses the big wave factor to filter out all market conditions with minor fluctuations, reducing transactional costs and improving the success rate of transactions.

Chapter 5: Conclusion

Genetic algorithm is a factor research method of “from Function to Logic”. Genetic algorithm may provide more possibilities for stock selection factor research. Although the content of the big wave factors studied in this research is complicated to explain with common logic, from the test results, the composition content of the big wave factor may have certain relationship with the stock price fluctuation and the lowest point of the daily stock price.

Big Wave Factor uses high-frequency data for factor mining, which is naturally more sensitive than traditional trend-following strategies and can quickly capture market conditions. In addition, the big wave factor captures the time point when there may be significant fluctuations, which can effectively exclude minor market conditions, reduce the number of transactions, reduce transaction costs, and increase the success rate of transactions.

This project provides a simple strategy for the big wave factor. Some parameters of this strategy deserve to be adjusted to obtain better returns. This compares such strategy with the traditional MACD moving average system strategy. It is found that this strategy has a significant advantage over the MACD strategy when the stock price fluctuates sharply, and the profits are not inferior in other situations. The big wave factor obtained an excess return of 0.8% during the three-day test, while the MACD strategy was only 0.3%. Besides, the income of the big wave factor is very stable.

Future work

Many details of the research method in this paper still need to be improved. For example, the algorithm can be optimized to speed up the training time; more data should be used for backtesting, and so on. These contents will be followed up in subsequent research. Although this article only provides three days of backtest data, this factor has achieved excellent results in the actual test through cooperation with an investment institution. In August 2021, the a-share liquor sector plummeted. This factor accurately warned of the intraday plummet point and accurately performed selling operations, reducing the institution's losses by 20%. Unfortunately, the a-share market does not support short selling, which limits the profitability of this factor. In addition, we are trying to delete part of the content of the factor without affecting its function, thereby increasing the analyzability of the characteristic and producing an element with an underlying logic that is not easy to fail.

References

1. Hendershott, T., C.M. Jones, and A.J. Menkveld, *Does Algorithmic Trading Improve Liquidity?* Social Science Electronic Publishing.
2. Jarnecic, E., *An Analysis of Trades by High Frequency Participants on the London Stock Exchange.* 2010.
3. Graham, B. and D.L. Dodd, *Security analysis*. 1934, New York,: Whittlesey house, McGraw-Hill book company, inc. xi, 725 p.
4. Basu, S., *Investment Performance of Common Stocks in Relation to Their Price-Earnings Ratios: A Test of the Efficient Market Hypothesis.* The Journal of Finance, 1977. **32**(3): p. 663-682.
5. Stattman, D., *Book Values and Stock Returns.* 1980.
6. Rosenberg, B., K. Reid, and R. Lanstein, *Persuasive evidence of market inefficiency.* The Journal of Portfolio Management, 1985. **11**(3): p. 9-16.
7. Marshall, J., *Snowball: Warren Buffett and the Business of Life.* 2008.
8. Covel, et al., *Trend following: how great traders make millions in up or down markets.* Futures, 2004(Jan).
9. Huang, M.Y., R.R. Rojas, and P.D. Convery, *Forecasting stock market movements using Google Trend searches.* Empirical Economics, 2019. **59**(6): p. 2821-2839.
10. Pavlova, I. and A.M. Parhizgari, *INTERNATIONAL MOMENTUM STRATEGIES: A GENETIC ALGORITHM APPROACH.* General Information, 1998.
11. Lee, C. and B. Swaminathan, *Price Momentum and Trading Volume.* Journal of Finance, 2000. **55**(5): p. 2017-2069.
12. COVEL and W. Michael, *COMPLETE TURTLE TRADER, THE - THE LEGEND, THE LESSONS, THE RESULTS.* 2009.
13. Daniel, K., D. Hirshleifer, and A. Subrahmanyam, *Investor Psychology and Security Market Under- and Overreactions.* The Journal of Finance, 2002.
14. Milnor, J.W. and G.A. Randall, *The Newfoundland-Azores High-Speed Duplex Cable.* Transactions of the American Institute of Electrical Engineers, 1931. **50**(2): p. 389-396.
15. Brogaard, J., et al., *High-Frequency Trading and Extreme Price Movements.* Journal of Financial Economics, 2018. **128**(2): p. 253-265.
16. Sezer, O.B. and A.M. Ozbayoglu, *Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach.* Applied Soft Computing, 2018: p. S1568494618302151.
17. Levendovszky, J., et al., *Low Complexity Algorithmic Trading by Feedforward Neural Networks.* Computational Economics, 2017.
18. Kissell, R., *The Science of Algorithmic Trading and Portfolio Management.* Elsevier Monographs, 2013.

19. Zuckerman, G., *The man who solved the market*. 2019, New York, NY: Portfolio / Penguin. xx, 359 pages, 8 unnumbered pages of plates.

20. Murphy, J.J., *Technical analysis of the financial markets : a comprehensive guide to trading methods and applications.* New York Institute of Finance, 1999.

21. Ozbayoglu, A.M., M.U. Gudelek, and O.B. Sezer, *Deep learning for financial applications : A survey.* Papers, 2020.

22. Karaoglu, S., U. Arpaci, and S. Ayvaz, *A Deep Learning Approach for Optimization of Systematic Signal Detection in Financial Trading Systems with Big Data.* International Journal of Intelligent Systems and Applications in Engineering, 2017. **SpecialIssue**(SpecialIssue): p. 31-36.

23. Liu, S., C. Zhang, and J. Ma, *CNN-LSTM Neural Network Model for Quantitative Strategy Analysis in Stock Markets.* Springer, Cham, 2017.

24. Gudelek, M.U., S.A. Boluk, and A.M. Ozbayoglu. *A deep learning based stock trading model with 2-D CNN trend detection*. in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2017.

25. Gunduz, H., Y. Yaslan, and Z. Ca Taltepe, *Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations.* Knowledge-Based Systems, 2017: p. S0950705117304252.

26. Sezer, O.B. and A.M. Ozbayoglu, *Financial Trading Model with Stock Bar Chart Image Time Series with Deep Convolutional Neural Networks.* 2019.

27. Thanh, D.T., et al., *Tensor Representation in High-Frequency Financial Data for Price Change Prediction.* Papers, 2017: p. 1-7.

28. Yue, D., et al., *Deep Direct Reinforcement Learning for Financial Signal Representation and Trading.* IEEE Transactions on Neural Networks and Learning Systems, 2016. **28**(3): p. 1-12.

29. Holland, J.H., *Genetic Algorithms and Adaptation.* Springer US, 1984.

30. Manahov, V., *The rise of the machines in commodities markets: new evidence obtained using Strongly Typed Genetic Programming.* Annals of Operations Research, 2016. **260**(1-2): p. 321-352.

31. Hauser, F., J. Huber, and B. Kaempff, *Costly Information in Markets with Heterogeneous Agents: A Model with Genetic Programming.* Computational Economics, 2014. **46**(2): p. 205-229.

32. Vasilakis, G.A., et al., *A Genetic Programming Approach for EUR/USD Exchange Rate Forecasting and Trading.* Computational Economics, 2012. **42**(4): p. 415-431.

33. Wang, L., et al., *Generating Moving Average Trading Rules on the Oil Futures Market with Genetic Algorithms.* Mathematical Problems in Engineering, 2014. **2014**: p. 1-10.

34. Miles, S. and B. Smith, *Testing The Adaptive Efficiency Of U.S. Stock Markets: A Genetic Programming Approach.* Journal of Business & Economics Research (JBER), 2010. **8**(11).

35. Yang, S.Y., et al., *Genetic programming optimization for a sentiment feedback strength based trading strategy.* Neurocomputing, 2017. **Online**(nov.15): p. 29-41.

36. Bauer, R.J. and F. Gregory Fitz-Gerald, *Using genetic programming to design a generalized trading system.* Managerial Finance, 2000. **26**(6): p. 1-15.
37. Akhtar, M.S., et al. *A Multilayer Perceptron based Ensemble Technique for Fine-grained Financial Sentiment Analysis*. in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017.
38. Dos, L., S. Pinheiro, and M. Dras, *stock market prediction with deep learning: a character-based neural language model for event-based trading.* 2019.
39. Zhang, J. and D. Maringer, *Using a Genetic Algorithm to Improve Recurrent Reinforcement Learning for Equity Trading.* Computational Economics, 2015. **47**(4): p. 551-567.
40. Kuo, R., C.H. Chen, and Y.C. Hwang, *An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network.* Fuzzy Sets and Systems, 2001.
41. Hu, Z., et al., *Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction.* 2017.
42. Vargas, M.R., B. Lima, and A.G. Evsukoff. *Deep learning for stock market prediction from financial news articles*. in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*. 2017.
43. Darwin, C., *On the Origin of Species.* Soil Science, 1915. **71**(6).
44. Alata, B. and E. Akin, *An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules.* Soft Computing, 2006. **10**(3): p. 230-237.