

# Build Robo-Advisor: A new method to coding K-line series

Fan Peiran<sup>1</sup>

*(Department of Quantitative and Technical Economics, University of Chinese Academy of Social Sciences, Beijing 102488, China)*

E-mail: [fanpeiran@ucass.edu.cn](mailto:fanpeiran@ucass.edu.cn)

**Abstract:** In the age of Big Data, Financial markets worldwide accumulate tones of data; we need a new method to build a frame so that new techniques like machine learning could have a chance reshape the foundation of the future market. This paper proposes a new frame to take the small step to this big future. This paper presented a new architecture of comprehensive system to define similarity could capture the almost every promising K-line patterns of the stock price with the power of big data techniques, this system allows us to generate patters, predict the possibility of correlated events attached, it is a necessary component in the field of finance technology. With this new tool, we predict direction of stock prices as a test. In order to define the K-line similarity, first we proposed a new coding system to every possible shape of K-line; then we coding the series of K-lines, with the decoding technique and inference approach algorithm we could find the transfer possibility of every possible pattern. Possibility turns to be the “knowledge” of this system. Naturally,” ARIMA with GARCH effect” model and Naïve Predict Model were chosen as the benchmark to test the feasibility of the system. Many specific patterns were thought to be magic in revealing the future of asserts, and the very shapes or patterns were based on experience of experts. In this study I use the data from Chinese stock market, by setting a series of basic method as benchmark. Many evidences show massive patterns search method based on K-line similarity match should be promising, the general predict power of pattern search is better using in massive predict rather than a single assert prediction. This tool is a new path to study patterns and events from K-line series, give a complete frame to train the deep learning networks like Generative Adversarial Network. By using this frame well, “Technical Analysis” will be reshaped.

**Key words:** Similarity, Machine Learning, K-line, Data Mining, GARCH, Naïve Prediction

---

<sup>1</sup> Fan Peiran, Department of Quantitative and Technical Economics, University of Chinese Academy of Social Sciences. [fanpeiran@ucass.edu.cn](mailto:fanpeiran@ucass.edu.cn)  
This work is protected by Chinese software intellectual property rights (Registration No. 2018SR702699).

## Introductions

The problem which we are concerned with is how to predict the price of stocks, building portfolios in the machine learning ways. In classical economic theory, if the market is efficient, any information will be no help to predict the future price of asserts, neither the “Fundamental Analysis” nor the “Technical Analysis”. This theory is called Efficient Markets Hypothesis (Fama, 1970), no one have the chance to “Beat the market, market knows more”. In the level of practice, almost every participates believe the market is not as efficient as the EMH theory claimed. “Technical analysis” is a set of techniques based on the K-line or some specific combination of the K-line series, the shape of the price itself will reveal its possible trend in the future, like a magic. In a word, history will recur. Actually the predictive power of K-line patterns is not concluded at least be limited by many conditions, thanks to the leap of computer science, big data techniques could search wider range of possibly patterns, with the massive pattern recognition methods and proper decoding techniques we could test the potential of the K-line patterns.

A typical architecture of pattern recognition system contains several common components: codification modules, pattern database, pattern match algorithm, decoding patterns and practice in the real world. Nowadays, most researcher have several ways using the toolbox in machine learning. Treat the K-lines as several time series is the main treatment, so the machine learning method like LSTM or CNN could be applied.( Taewook Kim, Ha Young Kim, 2019) Another alternative is using image retrieval technology also a solution to handle certain patterns, so that using data mining techniques such as K-mean Cluster method, researcher could find the perspective of small group of specific patterns.( Lv Tao, Yongtao Hao, el.al., 2017 and Leszek J Chmielewski, Maciej Janowicz, el.al., 2016) Traditional “Technical Analysis” usually use small group of patterns, but with the machine learning method we could investigate more patterns if we could have a general frame which could coding all

K-line series flexibly.(Nikitas Goumatianos, Ioannis Christou, et.al., 2013 and C.-F. Tsai, Z.-Y. Quan 2014) We focus on the access from K-lines to “patterns and events” which could be understood by the machine learning.

The K-line patterns is the central task of “Technical Analysis” (Nison,1991), traditionally it’s based human experiences, this paper propose a method to change status quo. In order to turn K-line into full scale test, we need coding K-line and K-line series, however, in the literature there are few papers focusing on the coding and decoding matters. After coding the features of K-line, we define the similarity method to match K-line series, we try to find out the future prospective of the same pattern, and eventually we could use the knowledge to forecast any K-line series.

In this paper we choose the Autoregressive Distributed Lag Model (ADL) as the rival model of K-line similarity match method. Naturally, as the heteroscedasticity problem, we use the Generalize Autoregressive Conditional Heteroscedastic Model to capture the volatility characteristics of the stock price (Engle, Robert, 1982). Autoregressive Moving Average Mode (ARMA) powered by GARCH is a powerful model to capture the dynamic structure of the data, also the predictive power make ARMA as a good academia reputation in the field of time series, that is the main reason we choose it as the benchmark. We also issue another predict method as the benchmark of both GARCH and K-line method, which is called Naïve Predict Model which take today’s condition as the prediction of tomorrow, in the end, we shall discuss the performance of three methods and give suggestions.

## **Data and Method**

In this paper we investigate the Chinese A shares price from 1990-12-19 to 2017-02-24, about 6998 trading days and 2972 stocks and 416 indexes of market, each assert carry the necessary data such as Open price, High price, Volume price et cetera. All the data as an open data is downloaded from WIND Database. We use 90% of data as the train set data to building the models, and the rest 10% of data is out samples which means 5762 trading days involved in the modeling sector. We use 500

trading days in 10% samples as the test set.

In this paper we will build a new kind of K-line similarity match method, and assess the performance of this model, and try to give the leads to refine this method. Firstly, some definitions are given in order to build this very method. Secondly, we use this model to predict the single assert price in multiple times, here we choose the 500 steps forth perspective of Shanghai Securities Composite Index (SSEC) as the study objects, and ARIMA/GARCH model and Naïve Predict Model is two rival models in this section. Next, we investigate the massive asserts one step forth using the K-line similarity match method, and Naïve Predict Model is chosen to be the alternative model.

### **Questions and Assumptions**

We draw series of questions and assumptions to investigate both the advantage and disadvantage of Similarity K line method. EMH is a big barrier of this paper, if it works in Chinese markets, all the frame we build is just build the tower in the sand. Fortunately, At least one method has more than 50% accuracy of prediction power in Chinese markets (as shown in the result section of this paper), so we can proceed our further test. In Chinese Stock market, short one stock is not so easy by the regulations, so we take the accuracy of “signals indicating rise” as the measure of the predicting power.

#### **Question 1: Is Similarity K-line method good enough in predicting the single assert compare another 2 models?**

We consider three possible ranking relations between them. We consider the performance as the judging evidence in this test. If K-line Similarity method is not the worst method, we think it has the potential to refine and research furthermore.

**H0:** K-line Similarity method is the **worst** method in three models in predicting the single assert.

**H1:** K-line Similarity method is the **best** method in three models in predicting the single assert?

**H2:** K-line Similarity method is the **moderate** method in three models in predicting the single assert?

If the performance of K-line Similarity method is not the worst, we will reject **H0**, and K-line Similarity method is good enough in the investing field.

As we know build the GARCH model to predict different asserts is a tedious and low efficient work, consider the perimeters training, deploying GARCH models to do massive prediction is not practical. We must understand massive predictions is not the same as single assert prediction, massive predictions require the models have characteristics such as flexible, robust and easy to train. Massive prediction still be a real problem in financial practice, so we choose Naive Predict as the rival model of Similarity K line method. When we say massive prediction in this paper, we try to predict the rise and fall situation of all 2972 stocks. The last question is given naturally.

**Question 2: Does Similarity K-line method be a good choice in predicting the massive asserts?**

For the same reason above, we take the accuracy of “signals indicate Rise” as the measure of the predicting power. There are two possible relationships between two methods.

**H3:** Similarity K-line method is better than Naive Predict in predicting the massive asserts.

**H4:** Similarity K-line method is worse than Naive Predict in predicting the massive asserts.

If we could answer these two questions properly, we could have the full understanding of the K-line similarity method which is derivative from our new frame. Then we may have a better way to study patterns and events from K-line series in the era of the Big Data.

### **Methodology of Constructing Similarity of K-line series**

In order to investigate all possible pattern of the shape of asserts, we need coding

the data, and constructing a well-defined similarity function. Firstly, we classified one K line as a start. A K-line bar carry information such as High price (H), Low price (L), Open price(O) and Low price(L), as shown in Table 1.

**Table 1:** Single line with 4 Prices (OHLC)

Symbol	Definition
H	Highest price of the trading day
L	Lowest price of the trading day
O	Open price of the trading day
C	Close price of the trading day

Now we could figure out every possible K-line, K-line like H\_C\_OL means Open price is equal to the Low price, Close price is in the middle, and High price is bigger than Close price, and Close Price is bigger than Open Price, in convenient it turns to be:

$$H > C > (O = L) .$$



**Figure 1:** All possible shape of single K-line

We deploy a two necessary method to describe the K-lines, first we call it “**Shape Code**” which concern the shape of every single K-line; the other we call it “**Position Code**” which describe the connections between K-lines. Then we give every possible shape of a single K-line a two digits number, like OHCL is resigned to be **Shape Code "00"**, which means all prices are equal. The more specific single K-line coding details could be found in the Table 2 and Figure 1.














**Table 2:** Single K-line coding Dictionary (Shape Code)

Code	Symbol	Definition
00	$OHCL$	$O = H = C = L$
01	$COH_L$	$(O = H = C) > L$
02	$OH_{CL}$	$(O = H) > (C = L)$
03	$OH_C_L$	$(O = H) > C > L$
04	$CH_O_L$	$(C = H) > O > L$
05	$H_{CO}_L$	$H > (C = O) > L$
06	$H_O_{CL}$	$H > O > (C = L)$
07	$H_C_O_L$	$H > C > O > L$
08	$H_O_C_L$	$H > O > C > L$
09	$CH_{OL}$	$(C = H) > (O = L)$
10	$H_{OCL}$	$H > (C = O = L)$
11	$H_C_{OL}$	$H > C > (O = L)$

Series of K-line could construct the patterns, so next, we try to coding the K-line series from study the position of two K-lines, actually if we could handle the relevant positions of two K-lines, we also could handle more K-lines as we want, it's just the matter of computing power. Naturally, we develop the “**Position Code**”. Idea is simple, we treat the link of two K-lines in a simple way. One K-line have two edges in space which determine the space they hold, top edge is “High”, naturally, ”Low price” makes a down edge, so one K-line turns to be a vector of two elements to present its future. Imagine two K line like two sticks move relatively, one move up, one down, every unique position is what we should concern. By this treatment we could investigate all the possible positions of two K-lines (13 possible positions, as shown in Table 3). It is also alright to choose “Open” and “Close” as the two dimensions to capture the feature of the K-lines, the important things we reduce the complexity of the problem and have the chance to capture the essence of the patterns.

We propose two coding techniques, one focus on the shape, the other focus on the position feature, according to the feature fusion system, we could catch the most features of the K-line series in an efficient way. Suppose we have 2 K-line, first K-line with High price  $H1$ , Low price  $L1$ , we represent  $(H1, L1)$  and for the same reason other K line is  $(H2, L2)$ , now we develop another coding system to represent the link of 2 K-lines. With this method we could represent the K line series no matter how long it is. The idea is simple, to a K-line series with the length of  $n$ , we could use this method  $C_n^2$  times, this recursion is the key to define the specific K-line series. Then we could match the similarity of any two kind of series with identical length.

**Table 3:** Coding dictionary for every possible position of 2 K-lines (Position Code)

Code	Symbol	Definition
01		$L1 > H2$
02		$L1 = H2$
03		$H1 > H2, L1 \in (L2, H2)$
04		$H1 > H2, L1 = L2$
05		$H1 > H2, L1 < L2$
06		$H1 = H2, L1 > L2$
07		$H1 = H2, L1 = L2$
08		$H1 = H2, L1 < L2$
09		$H1 < H2, L1 > L2$
10		$H1 < H2, L1 = L2$
11		$H1 < H2, L1 < L2$
12		$H1 = L2$
13		$H1 < L2$

Therefore, we give an example of 3 K-lines to show how we translate a K-line



series to one code with both two coding dictionaries. Naturally, combined “Shape Code” and “Position Code”, we define the first “K-line Similarity Method” in this new frame.

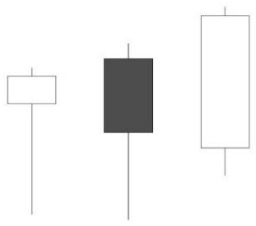
**“K-line Similarity Method”**: In this paper we define two K-line series which share the same “Shape Code” and “Position Code”, they are similar. In fact, in this frame we could add more dimensions of features as we wish, like “Volume”, “Market Value”, the idea is the same. K-line Similarity code  $\Psi^N$  can be defined as a code containing two dimensions, one is “Shape Code”  $V^N$  and the other is “Position Code”  $\Lambda^N$ , define Similarity  $\Psi^N = (V^N, \Lambda^N)$  as:

$$\Psi^N = \begin{cases} (V^N, \Lambda^N), & N \geq 2 \\ V^N, & N = 1 \end{cases}$$

For two sequences ( $s_i$  and  $s_j$ ) of equal length which length are  $N \geq 2$ , K-line Similarity code is  $\Psi_{s_i}^N = (V_{s_i}^N, \Lambda_{s_i}^N)$ ,  $\Psi_{s_j}^N = (V_{s_j}^N, \Lambda_{s_j}^N)$  respectively, if  $\Psi_{s_i}^N = \Psi_{s_j}^N$ , then two series is similar, and we have  $V_{s_i}^N = V_{s_j}^N$ ,  $\Lambda_{s_i}^N = \Lambda_{s_j}^N$ .

We define  $n$  K-lines similarity as “Shape Code” length of  $n$ , “Position Code” length of  $C_n^2$ . All patterns under this definition we have  $12^n \times 13^{C_n^2}$  unique pattern of K-lines similarity. When  $n = 3$ , we have 1728 different Shape Codes ( $12^3$ ), 2197 different Position Codes ( $13^{C_3^2}$ ), which construct 3796416 patterns of K-line similarity ( $12^3 \times 13^{C_3^2}$ ). When  $n = 4$ , we get 100088711424 unique patterns ( $12^4 \times 13^{C_4^2}$ ). For the convenient, in this paper, we introduce 3 K-lines similarity to predict.

**Table 4:** “Shape Code” and “Position Code” Construct the K-line Similarity ( $n = 3$ )

Method	Code	Example:
		A K-line series with 3 K-lines
<b>Shape Code</b> $V^3$	"080708"	
<b>Position Code</b> $\Lambda^3$	"110303"	
<b>Similarity Code</b> $\Psi^3$	(080708, 110303)	

Now we could build the model to match the similarity of K-line series, here is

some definition:

(I). Suppose we have a Euclidean space  $\Omega$  which contain lots of sequences with  $N$  K-lines, and events derivate by these K-lines.

$$\Omega = (\theta_1, \dots, \theta_M, E_1, \dots, E_J), \quad M, J \text{ is in the set of natural numbers.}$$

$$S_N^T \subseteq \Omega, S_N^T = (s_{1T}, \dots, s_{nT})$$

Then  $S_N^T$  is unique series which length is  $N$  in  $\Omega$  at time  $T - N$

to time  $T$ , which contain  $n$  unique  $s_{iT}$  in  $[T - N, T]$ . Naturally,  $S_N^{T+N}$  is the succeed states of  $S_N^T$ . We stack every  $\{S_N^1, \dots, S_N^T\}$ , and forge the element, we got a unique state dictionary, which contain every single unique patterns of length  $N$  K-lines.

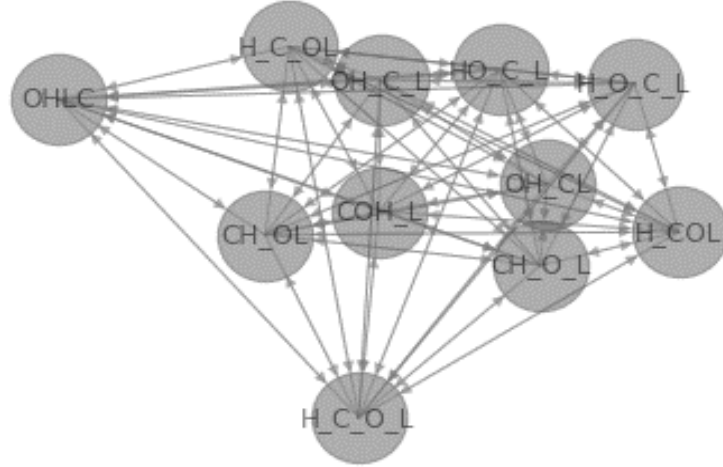
*Define StateDictionary:  $SD_N = \{v_1, v_2, v_3, \dots, v_n\}$ , as time goes,  $s_j$  growes too.*

Of course, we should update this “State Dictionary”, make sure its knowledge is latest.

(II). Define  $f(x|c) = \begin{cases} 1, & x \subseteq c \\ 0, & x \text{ not in } c \end{cases}$  ; *efine pattern pair*  $(\theta_{it}, \theta_{it+N}^*) \subseteq \Omega$ ,  $\theta_{it+N}^*$  consider to be the successive patterns of  $\theta_{it}$ . The transfer probability of pattern  $v_j$  To pattern  $v_k$  turns to be:

$$Prob(v_j \text{ to } v_k | \Omega) = \sum_i^M f(\theta_{it+N}^* | v_k) / \sum_i^M f(\theta_{it} | v_j),$$

When  $N = 1$ , we have the transfer matrix according “Shape Code”, as shown in Figure 2 and 3. We even find pattern which “shape code” is "07" have 87% probability transferred to itself.



**Figure 2:** Transfer Net between K-line types  
(Based on the Chinese stock market since 1992 to 2018)

	OHLC	COH_L	OH_CL	OH_C_L	CH_O_L	H_O_C_L	HO_C_L	H_C_O_L	H_O_C_L	CH_OL	H_COL	H_C_OL
OHLC	0.57	0.07	0.01	0.09	0.04	0.00	0.01	0.04	0.04	0.02	0.03	0.06
COH_L	0.03	0.04	0.03	0.12	0.12	0.00	0.04	0.30	0.27	0.01	0.00	0.05
OH_CL	0.01	0.00	0.02	0.08	0.04	0.00	0.06	0.40	0.26	0.01	0.02	0.10
OH_C_L	0.00	0.00	0.01	0.10	0.04	0.00	0.03	0.42	0.30	0.01	0.00	0.09
CH_O_L	0.00	0.01	0.01	0.09	0.07	0.00	0.03	0.39	0.33	0.01	0.00	0.06
H_O_C_L	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HO_C_L	0.00	0.00	0.02	0.07	0.04	0.00	0.06	0.40	0.29	0.01	0.01	0.09
H_C_O_L	0.00	0.00	0.00	0.02	0.01	0.00	0.01	0.87	0.08	0.00	0.00	0.01
H_O_C_L	0.00	0.00	0.01	0.08	0.03	0.00	0.02	0.43	0.33	0.01	0.00	0.08
CH_OL	0.02	0.01	0.01	0.09	0.09	0.00	0.02	0.35	0.33	0.02	0.00	0.06
H_COL	0.04	0.00	0.03	0.06	0.06	0.00	0.14	0.25	0.24	0.04	0.04	0.10
H_C_OL	0.00	0.00	0.01	0.09	0.03	0.00	0.02	0.41	0.34	0.01	0.00	0.09

**Figure 3:** Transfer matrix all patterns of single K-line (Shape Code)  
(Based on the Chinese stock market since 1992 to 2018)

(III). Define the pair to describe the process from  $\theta_{it}$  to certain event coming forth.

Define pattern pair  $(\theta_{it}, E_{it+a}^*)$ ,  $\theta_{it} \subseteq \Omega$ ,  $E_{it+a}^*$  consider to be the successive event of  $\theta_{it}$  in  $a$  steps forward.

The transfer probability of pattern  $v_j$  To pattern  $v_k$  turns to be:

$$Prob(v_j \text{ to } E | \Omega) = \frac{\sum_i^J f(E_{it+a}^* | E)}{\sum_i^M f(\theta_{it} | v_j)},$$

This ensure us to find out the connection of pattern to certain events.

**(IV):** When we investigate the perspective event we want the expected possibility converge to the decision: Positive or Negative , True or False, it is rooted the “belief” to every single pattern of this system, which should update the “belief” as data grows (empowered by the Bayes’s theorem). But for continent, in this primary study which main course is introducing this new method, we set just one threshold value for every event or pattern.  $\omega = 0.5$ ,

$$Prob(E|v_j) = \begin{cases} 1, & Prob(v_j \text{ to } E | \Omega) > \omega \\ 0, & Prob(v_j \text{ to } E | \Omega) \leq \omega \end{cases}$$

For example, suppose  $N = 3$ ,  $E$  is the rise or fall of the price 1 days later,  $v_j$  suppose to be the “Position Code” equals "110303", now we could match all the pattern in space  $\Omega$ , and find out the transfer probability of "110303" to the price ups and downs perspective one step forth. In this paper we define similarity in very simple way, when two K-line series have the same “Shape Code” and “Position Code” they share the same state in *StateDictionary*. And with help of (IV) we could predict the rise and fall of any asserts in the specific patterns.

### **ARIMA Model whit GARCH effect**

Predict the future price of a stock is typically a univariate time series problem, autoregression moving-average models is developed to handle such problems. *ARMA(P, Q)* is in the form:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \dots + \beta_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}$$

In this model  $y_t$  is dependent variables and the lag of  $y_t$  such as  $y_{t-1}, \dots, y_{t-p}$  is independent variables,  $\varepsilon_t$  and the lag of  $\varepsilon_t$  is also independent variables. If  $\varepsilon_t$  is not white noise, and the volatility clustering effect happens, we consider the Autoregressive Conditional Heteroskedasticity (ARCH), *ARCH(q)* is in the form (Engle 1982):

$$\sigma_t^2 \equiv Var(\varepsilon_t | \varepsilon_{t-1}, \dots, \varepsilon_{t-q})$$

If  $q$  is big enough, too much parameters in this high-order model will consuming too

much samples, gains so little information, so we consider using the autoregressive  $\sigma_t^2$ , this brilliant idea works well (Bollerslev 1986). The Generalized autoregressive conditional heteroscedastic model is an extension of ARCH model,  $GARCH(p, q)$  is the general form:

$$\sigma_t^2 = k + \gamma_1 \sigma_{t-1}^2 + \dots + \gamma_p \sigma_{p-1}^2 + \alpha_1 \varepsilon_{t-1}^2 + \dots + \alpha_q \varepsilon_{q-1}^2$$

Furthermore, the univariate time series better be stationary to ensure the ARMA model works well, or at least could turn to be the stationary series through some standard procedure which usually to use differencing techniques, this is called to be unit-root nonstationary problem. If we have unit-root, in order to handle unit-root problem, ARMA model turns to be ARIMA model. An  $ARIMA(P, 1, Q)$  process means  $y^* = y_t - y_{t-1} = (1 - B)y_{t-1}$ , which is an ARMA process. Cause we turns stock price as the subject, naturally, the daily return is stationary, so  $ARIMA(P, 0, Q)$  is just fine. Our prime purpose of modeling is find a bunch of parameters of the univariate time series model to capture the necessary features of the stock price, make sure the residuals of this model is just white noise which no more information be attached, finally we use this benchmark model to predict rise and fall of the stock price in the out sample times. Naturally, because this is a benchmark model, not our prime interest lies, many details of statistic techniques will be unnecessary, in one word, we do much exploring experiment to determine a simple, robust and effective parameter set.

## Test the predict power of models

### Forecast the single stock price with K-line pattern cognition

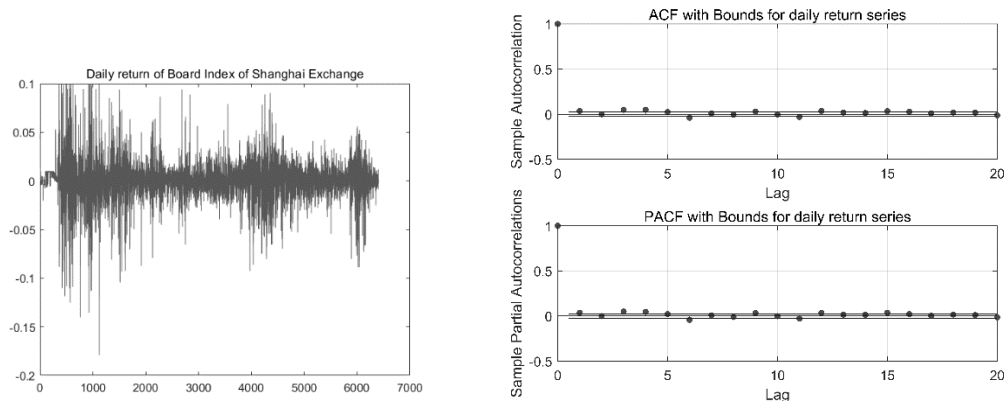
We choose Shanghai Securities Composite Index (SSEC) as our target to test the performance of three models.

We use daily return as our objective in this model,  $r_t$  generate from Close price,

$$r_t = (Close_t - Close_{t-1})/Close_t.$$

We use three models investigate the daily return of Index of Shanghai Exchange, by

the single asset predict performance of three models we could investigate the K-line method objectively. Firstly, we build a *ARIMA* model with *GARCH* effect using the daily data of Shanghai Securities Composite Index.



**Figure 4:** Daily return series and ACF and PCF

(Based on the data of Shanghai Securities Composite Index (SSEC) )

Then we got two equations to describe the daily return  $r_t$ :

$$\begin{aligned}
 r_t = & 0.7796r_{t-2} - 0.3017r_{t-6} + 0.1719r_{t-7} - 0.1845r_{t-13} + 0.3438r_{t-15} + \epsilon_t \\
 & + 0.0600\hat{\epsilon}_{t-1} - 0.7689\hat{\epsilon}_{t-2} + 0.2534\hat{\epsilon}_{t-6} - 0.1390\hat{\epsilon}_{t-7} \\
 & + 0.0460\hat{\epsilon}_{t-8} + 0.2370\hat{\epsilon}_{t-13} - 0.3461\hat{\epsilon}_{t-15} \\
 \hat{\sigma}_t^2 = & 3.7849 \times 10^{-6} + 0.88557\hat{\sigma}_{t-1}^2 + 0.11219\hat{\epsilon}_{t-1}^2
 \end{aligned}$$

beware  $\hat{\epsilon}_t$  and  $\hat{\sigma}_t^2$  is not the real value but the estimated residual value by this model, with these equations we could processing the prediction, more statistic details of this model presented in the table below.

**Table5:** *ARIMA*(15,0,1)/*GARCH*(1,1) Estimation (*ARIMA*( $P, D, Q$ ) part)

Parameter	Value	Standard Error	T Statistic
<i>AR</i> (2))	0.7796 ***	0.07189	10.844
<i>AR</i> (6)	-0.3017 ***	0.053415	-5.6491
<i>AR</i> (7)	0.1719 ***	0.056359	3.0507
<i>AR</i> (13))	-0.1845 ***	0.087631	-2.1053
<i>AR</i> (15)	0.3438 ***	0.081433	-2.1053
<i>MA</i> (1)	0.0598 ***	0.0089948	6.6441
<i>MA</i> (2)	-0.7688 ***	0.07282	-10.557

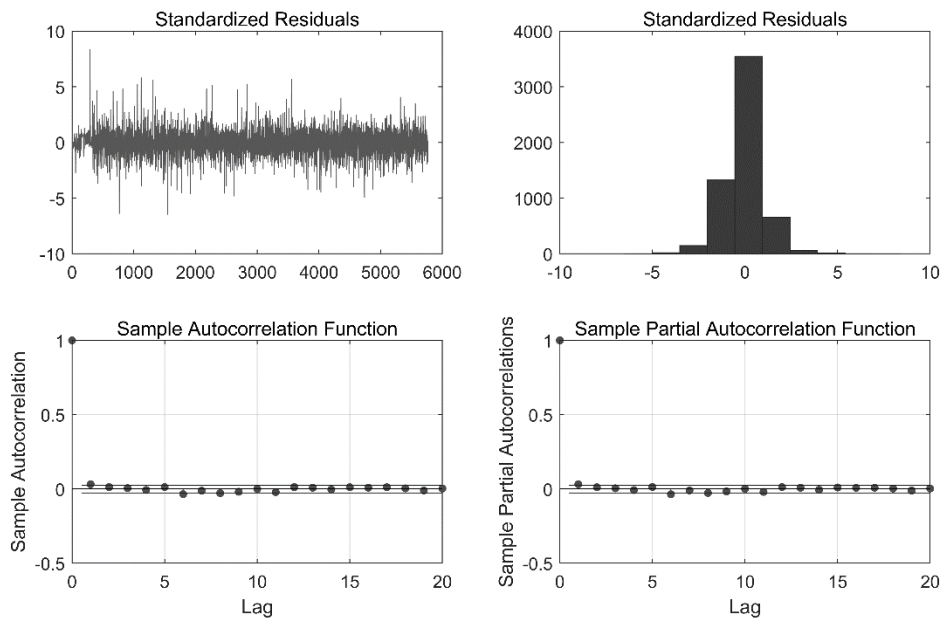
$MA(6)$	0.2534 ***	0.056518	4.4835
$MA(7)$	-0.1390 ***	0.053432	-2.6015
$MA(8)$	0.0461 ***	0.0155	2.9739
$MA(13)$	0.2370 ***	0.08898	2.6638
$MA(15)$	-0.3461 ***	0.082377	-4.2019

\*\*\*, \*\*, \*significant at 1%, 5% and 10%.

**Table 6:**  $ARIMA(15,0,1)/GARCH(1,1)$  Estimation ( $GARCH(p, q)$  part)

Parameter	Value	Standard Error	T Statistic
<i>Constant</i>	3.7e-06 ***	2.9772e-0	12.713
$GARCH(1)$	0.8856 ***	0.0033021	268.18
$ARCH(1)$	0.1122 ***	0.0039963	28.074

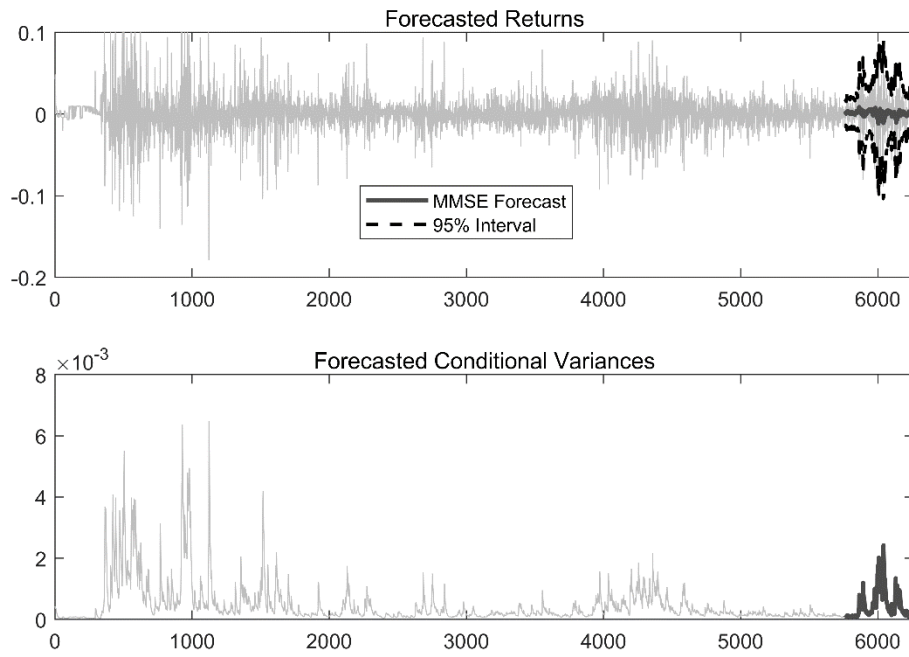
\*\*\*, \*\*, \*significant at 1%, 5% and 10%.



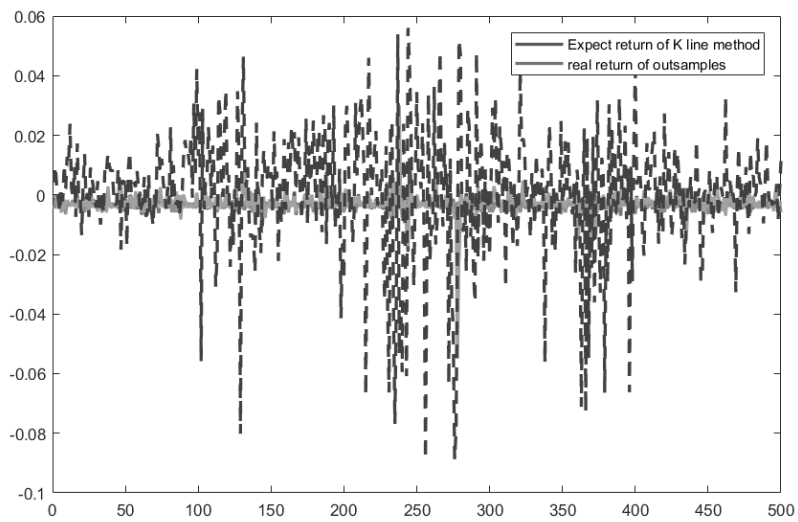
**Figure 5:** Residuals and ACF and PCF

When we capture the necessary information of the univariate time series, residuals turn to be random walk which be no help to predict the price of the stock. Then we using this model predict one step more, as we update the new data, we could predict the next 500 days rise and fall out samples. 63.4% seemed be pretty good, this very model out sample predict have 63.4% correct ratio while 36.6% of the result is

wrong.



**Figure 6:** Forecast performance 500 steps forward  
(Based on the  $ARIMA(15,0,1)/GARCH(1,1)$  model)



**Figure 7:** The predict performance K-line similarity match model  
(Based on the data of Shanghai Securities Composite Index (SSEC))



## Result of predict single assert in 500 steps forth

As shown from the Table 7, we could see, the K-line similarity method and *ARIMA/GARCH* have more than 50% accuracy of predict power, which means EMH is not the whole truth in the market of the Chinese A share stock markets, the try to predict the market is not wholly useless. We use the accuracy of “signals indicating Rise” as the major indicator to measure the performance of predicting power.

**Table 7:** The predict result of Shanghai Securities Composite Index

Method	Beat Ratio	Beat Ratio(rise)	Beat Ratio(fall)
<i>ARIMA/GARCH</i>	63.40%	<b>64.15%</b>	61.24%
Naive Predict	49.40%	56.10%	40.38%
K-line method	56.40%	59.07%	47.37%

As shown in Table 7, *ARIMA/GARCH* model have big advantage in predicting single assert, the signals of rise have 64.15% accuracy, while the Naive Predict got 56.10%, K-line method got 59.07%. So the answer of **Question 1** is obviously, in the field of single predict field K-line method is not as good as *ARIMA/GARCH* model, so technical researcher should this K-line carefully especially when they comment to the single assert. But we could all so see, Naïve Predict Method is an inferior method comparing to the K-line. K-line Similarity method is a moderate model in predicting the single assert (**H2**), but it still needs to be refined to catch up to the *ARIMA/GARCH* model.

## Massive Predict result of all A share stocks 2018

K-line method rise signal 119 times and 96 are correct, in Chinese Stock market, short one stock is not so easy by the regulations, so this signal really counts in investing practice. As shown in Table 8, we are pretty sure, when you want to predict the Chinese stock market and decide which assert to invest; K-line method is indeed a good option. As a matter of fact, the answer of **Question 2** is obviously, in the massive predict field K-line method is better than Naive Predict (**H3**).

**Table 8:** Massive Predict result of all A share stocks 2018

Method	Beat Ratio	Beat Ratio(rise)	Beat Ratio(fall)
Naive Predict	44.20%	51.78%	40.05%
K-line method	47.22%	<b>80.67%</b>	27.01%

## Conclusion

In this paper we give the full procedures to build the new frame to processing the K-line series, we combined the “Shape Code” and “Position Code” to a new similarity method. By this well-defined Similarity method could clustering the K-line patterns from different markets. With this new tool, the big data and machine learning techniques could be involved, so that “Technical Analysis” would be reformed in the new era. If the EMH is not the wholly truth of the markets, then we still have the chance to gather the useful information to predict which is the whole basis of investing activity. Fortunately, markets seem to be not always efficient.

In order to test the predict power of the K-line similarity method, we arrange two kind of test which are “single assert prediction” and “massive prediction”. The “single assert prediction” result shows this method is better than Naïve Prediction, but it is not as good as *ARIMA/GARCH*. Further study will be needed to refine the predict power of K-line similarity in this field.

The reason why *ARIMA/GARCH* could not be the benchmark of Massive prediction, answer is simple: cost. As known by every experienced researcher, build *ARIMA/GARCH* is time consuming. Choosing right lags for *ARIMA/GARCH* is tedious, and the training parameter set could not applicate to the other assert even to the same assert in different time intervals. If you use this method to predict every assert in the market tomorrow, it should be disastrous, the output of models may need days to give

you the answers even with a huge computer resource. In this paper we use 3 K-lines as the subject to mining the patterns, it is also because the cost concern. As this study argued, K-line similarity method does not have the advantage in the single assert prediction comparing with *ARIMA/GARCH* model. Generally, this new frame is worthy to deploying in massive prediction.

## Further Discussion

For convenient, we use 3 K-lines similarity to start this research. In the future we will deploy the  $n > 4$  K-lines similarity which could capture more features of the patterns of K-lines. In this paper, we use the fixed threshold to confirm the event occurrences; also, we treat transfer probability as the static parameter. In the future, we would update these parameters dynamically. We believe Bayes inference augmented K-line similarity method will be reframed, the predict power of this frame would grows as the data grows. We believe if refine properly the potential of this frame would be developed, as a matter of a fact, the way we define the similarity of the K-line series would have big potential.

As we argued before, K-line similarity is the necessary component for the finance technology, using this method. We could generate the K-line series which could train the human traders or big scale neural nets especially for the Generative Adversarial Networks (GANs) targeting in the field of financial investment. Also, we could use the data from markets worldwide with a single frame, that is the essence of this frame.

## Reference

Fama, Eugene. Efficient Capital Markets: A Review of Theory and Empirical Work[J]. *Journal of Finance*,1970 ,25(2).

Taewook Kim and Ha Young Kim.Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data[J]. *PLoS ONE*, 2019,14(2).

Lv Tao, Yongtao Hao, Hao Yijie, and Shen Chunfeng.K-Line Patterns' Predictive Power Analysis Using the Methods of Similarity Match and Clustering[J].*Mathematical Problems in Engineering*, 2017,2017(30).

Leszek J. Chmielewski, Maciej Janowicz, Arkadiusz Orłowski.Prediction of Trend Reversals in Stock Market by Classification of Japanese Candlesticks[J].*Advances in Intelligent Systems and Computing book series*,2016, 403.

Nikitas Goumatianos, Ioannis Christou and Peter Lindgren.Stock Selection System: Building Long/Short Portfolios Using Intraday Patterns[J]. *Procedia Economics and Finance*, 2013,2013.

C.-F. Tsai, Z.-Y. Quan.Stock prediction by searching for similarities in candlestick charts[J].*ACM Transactions on Management Information Systems*, 2014, 5.

Engle, Robert F. Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation[J].*Econometrica*,1982,50.

Box, G. E. P., G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. 3rd ed [M]. Englewood Cliffs,NJ: Prentice Hall,1994.