Predicting Chinese stock market price trend using machine learning approach

Chonagyang Zhang
Institution of Automation, Chinese Academy of Sciences,
No.95 Zhongguancun EastRoad, Haidian District, Beijing
China

University of Chinese Academy of Sciences

No.19(A) Yuquan Road, Shijingshan District, Beijing,

China

chongyang.zhang@ia.ac.cn

Zhi Ji
Institution of Automation, Chinese Academy of Sciences,
No.95 Zhongguancun EastRoad, Haidian District, Beijing
China

Zhiji.vip@gmail.com

Jixiang Zhang
Institution of Automation, Chinese Academy of Sciences,
No.95 Zhongguancun EastRoad, Haidian District, Beijing
China

jixiang.zhang@ia.ac.cn

Yanqing Wang
Institution of Automation, Chinese Academy of Sciences,
No.95 Zhongguancun EastRoad, Haidian District, Beijing
China
yanqing.wang@ia.ac.cn

Xingzhi Zhao
The Governor's Academy
Elm Street, Byfield, Massachusetts,
United States of America
Xingzhi.zhao@govsacademy.org

Yiping Yang
Institution of Automation, Chinese Academy of Sciences,
No.95 Zhongguancun EastRoad, Haidian District, Beijing
China
yipping.yang@ia.ac.cn

ABSTRACT

The stock market is dynamic, noisy and hard to predict. In this paper, we explored four machine learning models used technical indicators as input features to predict the price trend 30 days later. The experimental dataset is Shanghai Stock Exchange(SSE) 50 index stocks. The result demonstrates that ANN performs better than the other three models and is promising to find some profitable patterns.

KEYWORDS

machine learning, stock prediction, SVM, neural network,

CSAE 2018, October 22–24, 2018, Hohhot, China
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6512-3/18/10...$15.00
https://doi.org/10.1145/3207677.3207679

1 INTRODUCTION

Financial market is risky, chaotic, complex , dynamic, and full of uncertainties. Many factors such as economic policy, breaking news, political events, investors' sentiments may cause the fluctuations of asset market. Stock price analysis and prediction remain a challenging issue in both academic community and investment industry. With the increasing volume of financial data and growing computing power, meaningful information could be extracted from the market to help investors to make appropriate decisions, improve profitability and avoid potential loss. Forecasting stock market prices and trend plays an import role in financial decision making, investment management and algorithmic trading.

Traditional models used in stock prediction involved statistical methods such as time series model and multivariate analysis, which are often from a mathematical point of view. The financial value were considered as a function of time series and solved as a regression problem. The applications of machine learning approach in the stock market solve the problem in a new way. When viewing the problem as a classification problem, the result performance could be better [1-4]. Our goal is to design an intelligent machine learning model that learns from historical data and predict the future direction of the market. A variety of algorithms such as SVM, Neutral Network, Naive Bayesian Classifier, Random forest is explored in this paper. Most data source of the previous research is developed countries such as America, Japan. In this paper, we use Chinese stock market data to explore the effectiveness of machine learning models. Due to the difference of regulation law and social and political systems, Chines stock market may have its own characteristics. It is meaningful to explore these machines learning models in the Chinese stock market.

2 Related Work

The application of machine learning algorithm to predict stock price trends is contradictory with the Efficient Market Hypothesis[5]. The Efficient Market hypothesis demonstrates that all the information is reflected on the stock price such that it is impossible to predict the stock prices in the future.
However, the Efficient Market Hypothesis is highly controversial. Many researchers and investors believe that this theory may be correct in an ideal environment but not the real market in which different people may have different information priority, and traders could not always be rational[1-3, 6-9]. Trading algorithms could be successful in forecasting the dynamic and complex market.
Technical analysis is one of the mainstream analytical approaches for forecasting the directions of prices through history information, primarily past price and volume [10, 11]. Technical indicators could be derived from price, volume and time and are often used as input features to forecast financial series as a technical approach. A technical indicator, such as MACD, SMA, Relative Strength Index(RSI) is computed by a composition prices of open, low, high, close of a stock over a certain period time. For example, MA5 is average price over 5 days of a given stock. [6]compares ARIMA and Artificial Neural Network Models for stock price prediction and reveals the superiority of ANN model over ARIMA. [12] evaluated multiple predictions models and found that random forest perform best. [13] investigated different combinations of input window length and forecast horizon when calculating technical indicators using SVM. [4] applied machine learning techniques to predict the close price of the OMV Petrom stock and achieved encouraging results. [14] explored one-step and multi-step stock prediction using ANN. [15-16] mainly focused the application of Hidden Markov Model in the stock market prediction, and the stock price fluctuations were also analyzed.

3 Prediction System

3.1 Input Features

Ten technical indicators(TIs) are calculated to form the input feature vectors. Each indicator delivers information about the stock in a different point of view, such as the stock volatility, the strength or weakness of the trend and so on. TIs are calculated from the raw trading data which include open, high, low, close prices, trading volume and total money. Therefore, for a certain trading day, ten Tis could be derived from the raw trading time series data. These ten technical indicators are the same as used in [13]. Details about the ten TIs are listed below.

1 Simple Moving Average(SMA) is a trend indicator. It is the calculation result of the average price over the past n days

$$SMA_n = \frac{1}{n}\sum_{i=0}^{n-1} C_{t-i}$$

where C_t is the close price on day t and n is the input window length

2 Exponential Moving Average(EMA) is a type of moving average with weights decreasing exponentially, more weights are assigned with recent data, the weights sum up to 1,n is the input window length.

$$EMA_n = \frac{1}{n}\sum_{i=0}^{n-1} \omega_i C_{t-i}$$

3. Average True Range(ATR) depicts the degree of price volatility. It is computed as following

$$ATR_n = EMA_n(\max(H_t - L_t, | H_t - C_{t-1} |, | L_t - C_{t-1} |))$$

H_t, L_t, C_t are high, low, prices on day t, and n in the input window length.

4 Average Directional Movement Index (ADMI) indicates the strength of a trend regardless of whether it is up and down. DI_n^+ and DI_n^- ,calculate over a period of n past days corresponding to the input window length.

$$ADMI_n = 100*(DI_n^+ - DI_n^-)/(DI_n^+ + DI_n^-)$$

$$DI_n^+ = 100*EMA_n(DM^+)/ATR_n$$

$$DI_n^- = 100*EMA_n(DM^-)/ATR_n$$

where

$$DM^+ = \max(C_t - C_{t-1}, 0)$$

$$DM^- = \min(C_t - C_{t-1}, 0)$$

are positive and negative directional movements.

5 Commodity Channel Index(CCI) is an indicator that could be used to identify a new trend or warn extreme conditions.

$$CCI_n = (M^t - SMA_n(M^t))/(0.015\sum_{i=1}^{n} | M_{t-i+1} - SMA_n(M^t)/n |)$$

$$M^t = H_t + L_t + C_t$$

6 Price rate of change(ROC) shows the relative difference between the closing price on the day of forecast and the close price of n day previously, also could be considered as the percent of price change over the past n days.

$$ROC_n = (C_t - C_{t-n})/C_{t-n}$$

7 Relative Strength Index(RSI) is a momentum indicator that compares the magnitude of recent gains and losses during a certain time period. It could uncover the strength or weakness of a price trend.

$$RSI = 100 - 100/(1 + EMA_n(DM^+)/EMA_n(DM^-))$$

8 Williams %R is a momentum indicator measuring overbought and oversold levels. It is derived from the current closing price and the high and low prices over the latest n days equal to the input window length.

$$Williams\%R_n = 100*(H_n - C_t)/(H_n - L_n)$$

9 Stochastic %K is a technical momentum indicator gives information whether a stock is oversold or over bought by comparing a close of a stock and close prices time series during the past of n days.

$$\%K_n = 100*(C_t - LL_n)/(HH_n - LL_n)$$

10 Stochastic %D gives a turnaround signal meaning whether the stock is oversold or over bought. It is the calculation result of a

3-days EMA of Stochastic %K over a certain period of time defined by the input window length n.

$$\%D = EMA_3(\%K_n)$$

Input window length and forecast horizon are of great importance to calculating our ten TIs. The first parameter means how many past days of data we use to calculate the technical indicators. For example, when input window length is 5, SMA means average price of a stock over past 5 days, when the input window length is 10, SMA is the average price over past 10 days. Forecast horizon, also could be named as prediction step. It means how many days after we are going to predict. Most papers mainly predict the stock price of the next day. In this case, the forecast horizon equals to 1.[13]found that, when the input window length is approximately equal to the forecast horizon, the best prediction performance could be achieved. Considering that there could be more predictability in the long-term run, we choose input window length and forecast horizon both equal to 30.

3.2 Machine learning Models

SVM(support vector machine)

The main idea of SVM is to find the optimal classification hyper-plane that separate two classes of sample data with the best margin. Best margin means that the distance of the nearest samples of both classes to the hyper-plane achieves maximum. Those data points defined the separation plane are named as support vectors.

The standard model is as follows:

$$\begin{aligned} &\min(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i) \\ s.t.\quad & y_i(<w,\phi(x_i)> + b) + \xi_i \geq 1, i = 1,2,...l \\ & \xi_i \geq 0, i = 1,2,...l \end{aligned}$$

Where x is the input training data and y is the corresponding class label. ϕ is a nonlinear kernel function mapping from the original space to a high dimensional feature space. w,b are the weights of the model . C is a predefined parameter that trades off between the training accuracy and generalization ability

Neutral networks

The neural network model, also known as Artificial Neural Network(ANN) is one of the most popular artificial intelligence algorithms inspired by the human neural systems. Artificial neural network is essentially a directed graph consists of layers of neurons linked with each other with different weights. The input layer receives the input features, the output layer outputs the final result of processing. The layers in between are called hidden layer, the function of hidden layers could be considered as finding the appropriate nonlinear mapping such that the best classification result could be achieved.

Random Forest

Random forest is an ensemble learning method by constructing multiple trees at training time and outputting the final class label. Decision trees have very low bias and high variance. Small noise in the data could lead the tree grow in a completely different manner. This weakness is avoided in random forest by training multiple decision trees on different subspace of the feature space at the cost of slightly increased bias. None of the trees in the forest could see the entire training data. The data is recursively split into partitions. A particular node is built according to a certain attribute. The choice of the separating criterion is based on some impurity measures such as Shannon Entropy or Gini impurity.

Naive Bayesian Classifier

A naïve Bayes classifier is an algorithm to classify data based on the Bayes' theorem. This algorithm assumes that all attributes of a data point under consideration are independent with each other. It is easy to implement and efficient on large scale of data.

We use the prediction accuracy to evaluate all of the above models.

4 Experiments

4.1 Raw data

We choose Shanghai Stock Exchange (SSE) 50 index stocks as our experimental data. These 50 stocks represent top 50 stocks with large scale and good liquidity in the Shanghai stock market. The data is from 2010.01.01 to 2018.05.20. The choice of data range mainly depend on two factors, one is the data volume is big enough for training, the other is that we intend to use latest dataset. The raw data includes open price, close price, high price, low price and trading volume for the day.

We split the first 80% of data as training set, the remained 20% data as test set. There is no overlapping between the training set and test set, making sure that no future information is used to prediction the stock trend. The stock price data is provided by ricequant, a popular quantitative trading platform in China.

4.2 Preprocessing

Data preprocessing is an essential step in order to transform raw stock prices into a form acceptable for applying machine learning algorithms. The pre-processing steps are listed below.

Feature extraction of the original data into a set of TIs. Ten Tis are computed for each data point of each stock and used as input features of our machine learning models.

Normalization of data set is applied after feature extraction so that each input feature had zero mean and unit variance. The mean and variance are calculated based on the training dataset, then these values are applied to normalize both the training and testing dataset.

4.3 Data labelling

We classified the directions of future price movements into two classes. The assignment of the labels two each data sample is performed by comparing the close price of present day the close price n day after, n is the prediction step. The following equation illustrates the labelling method.

$$label = \begin{cases} 'up' & C_{t+s} > C_t \\ 'down' & C_{t+s} < C_t \end{cases}$$

Where s is forecast horizon. C_t and C_{t+s} are closing prices of a stock on day t and day t+s, respectively.

The label ‘up’ is assigned to a data point when the corresponding close price went up. If the corresponding closing stock data went down , the data point is labeled as ‘down’.

4.4 Experimental Results

This section provides some detail information about implementation of the algorithms and parameters tuning in the experiment.

The SVM model has hyper-parameters includes γ and a penalty rate for misclassification C ,both of these two parameters have significant affects to the performance of the model thus need to be optimized. The kernel function used in the SVM model is rbf. We applied grid search to identify good parameters combination. The values of γ and C are selected from exponentially growing sequences gamma=(2-15,2-13,...23)and C=(2-5,2-3,...215) respectively, as suggested in[13] . We employ five-fold cross-validation to seek optimal parameters of the SVM model. The whole training dataset is divided into five parts, among which four parts are used for training and the remained fifth part for testing. The procedure is repeated five times so that each part could be used for testing. The performance of different parameters settings is evaluated based on the prediction accuracy which is the percentage of the correct classified data point over the total number of data points. Once the optimal parameters are chosen, they are used to classify the testing data point.

The ANN models are implemented using Tensorflow, a popular deep learning framework offered by Google. Our feedforward neural network contained three layers. The input layer has ten input neurons to represent ten calculated Tis as input features. The number of neurons of hidden layers is set to ten for all stocks. The output layer has two neurons, indicating our classification result, whether the stock price is going up or down after the certain period of time defined by the prediction step. The activation function used in our neural network is relu. Random forest and naïve bayes are implemented using scikit-learn library.

The classification accuracy of different models are listed in table 1.We can observe that for every individual stock, the performance of the four machine learning models is quite different. The average prediction accuracy of 50 stocks is 53.9% for ANN, 48.6% for naïve bayes, 50.4% for random forest and 51.4% for SVM, showing that ANN is able to find some predictable patterns. SVM achieves the best average training accuracy 69.76%, however its average test accuracy is lower ANN model. One thing should be noted is that the stock 601878, the prediction accuracy of this stock for SVM and RF is 100%, and ANN only achieves 74.1% and NB for 85.2%. This is because all the labels in the dataset are down, during the period of test time, the price of the stock is always lower that the price of 30 days ago.

5 Conclusion and future directions

In this paper, we applied four well-known machine learning models to predict the rise and fall of a stock after 30 trade days. We evaluated the prediction accuracy of Shanghai Stock Exchange(SSE) 50 index stocks. The result demonstrates the superiority of ANN over the other three models. The financial market generates huge amount of data every day, our research shows that ANN may be a promising procedure in finding profitable patterns in the stock market. This paper only considers technical indicators as input , however, the macro-economic information is also very important and valuable in the prediction of the stock market. In the future research, we will incorporate macro-economic information into machine learning models and explore the application recurrent neural networks such as LSTM in the financial market analysis.

REFERENCES

[1] Bisoi, R. and P.K. Dash, A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter. Applied Soft Computing, 2014. 19: p. 41-56.

[2] Chang, T.S., A comparative study of artificial neural networks, and decision trees for digital game content stocks price prediction. Expert Systems with Applications, 2011. 38(12): p. 14846-14851.

[3] Chen, R.Y. and B. Pan, Chinese Stock Index Futures Price Fluctuation Analysis and Prediction Based on Complementary Ensemble Empirical Mode Decomposition. Mathematical Problems in Engineering, 2016.

[4] Cocianu, C.L. and H. Grigoryan, Machine Learning Techniques for Stock Market Prediction. A Case Study of Omv Petrom. Economic Computation and Economic Cybernetics Studies and Research, 2016. 50(3): p. 63-82.

[5] Fama, E., Efficient market hypothesis: A Review of Theory and Empirical Work. 1970.

[6] Adebiyi, A.A., A.O. Adewumi, and C.K. Ayo, Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. Journal of Applied Mathematics, 2014..

[7] Akita, R., et al. Deep learning for stock prediction using numerical and textual information. in Ieee/acis International Conference on Computer and Information Science. 2016.

[8] Singh, R. and S. Srivastava, Stock prediction using deep learning. Multimedia Tools and Applications, 2017. 76(18): p. 18569-18584.

[9] Sugumar, R., A. Rengarajan, and C. Jayakumar, A Technique to Stock Market Prediction Using Fuzzy Clustering and Artificial Neural Networks. Computing and Informatics, 2014. 33(5): p. 992-1024.

[10] Lahmiri, S., Entropy-Based Technical Analysis Indicators Selection for International Stock Markets Fluctuations Prediction Using Support Vector Machines. Fluctuation and Noise Letters, 2014. 13(2).

[11] Schumaker, R.P., et al., Prediction from regional angst - A study of NFL sentiment in Twitter using technical stock market charting. Decision Support Systems, 2017. 98: p. 80-88.

[12] Ballings, M., et al., *Evaluating multiple classifiers for stock price direction prediction.* Expert Systems with Applications, 2015. **42**(20): p. 7046-7056.

[13] Shynkevich, Y., et al., Forecasting Price Movements using Technical Indicators: Investigating the Impact of Varying Input Window Length. Neurocomputing, 2017.

[14] Dong, G., K. Fataliyev, and L. Wang. *One-step and multi-step ahead stock prediction using backpropagation neural networks.* in *Communications and Signal Processing*. 2014.

[15] Somani, Poonam, Shreyas Talele, and Suraj Sawant. "Stock market prediction using hidden Markov model." *Information Technology and Artificial Intelligence Conference (ITAIC), 2014 IEEE 7th Joint International.* IEEE, 2014

[16] Gupta, Aditya, and Bhuwan Dhingra. "Stock market prediction using hidden markov models." *Engineering and Systems (SCES), 2012 Students Conference on.* IEEE, 2012..

Table 1 experimental results of different models

	ANN		NB		Random Forest		SVM	
	TrainAcc	TestAcc	TrainAcc	TestAcc	TrainAcc	TestAcc	TrainAcc	TestAcc
600000	0.5675	0.4242	0.5154	0.3985	0.8387	0.6093	0.9325	0.6401
600016	0.6131	0.4267	0.6035	0.4550	0.8053	0.4499	0.8188	0.4396
600019	0.6420	0.4267	0.5771	0.5167	0.8753	0.2879	0.5289	0.3625
600028	0.6375	0.3393	0.6170	0.4062	0.8464	0.3445	0.6427	0.4344
600029	0.6568	0.4216	0.6093	0.3985	0.8580	0.4627	0.6337	0.3985
600030	0.6067	0.4833	0.5668	0.5990	0.8303	0.5553	0.6041	0.4961
600036	0.5893	0.6735	0.5720	0.6144	0.8380	0.3419	0.5045	0.3239
600048	0.6157	0.4936	0.5778	0.4190	0.8856	0.5578	0.5958	0.5321
600050	0.6735	0.7095	0.6060	0.7481	0.7873	0.6735	0.6607	0.6967
600104	0.6446	0.5578	0.5546	0.3111	0.8580	0.3111	0.5861	0.3728
600111	0.6665	0.5630	0.6671	0.5656	0.8323	0.6375	0.6446	0.5270
600309	0.5951	0.4859	0.5983	0.3368	0.8766	0.4344	0.6035	0.4807
600340	0.6433	0.5321	0.6491	0.6375	0.8072	0.6504	0.6748	0.4936
600518	0.5578	0.5450	0.5501	0.3676	0.8657	0.4319	0.7564	0.6427
600519	0.6272	0.7866	0.5508	0.1491	0.8168	0.1491	0.5668	0.8509
600547	0.6048	0.6015	0.6067	0.5810	0.8515	0.5733	0.5283	0.5578
600606	0.6138	0.6041	0.6324	0.4319	0.8618	0.5219	0.5913	0.6632
600837	0.5829	0.4036	0.5694	0.4679	0.8046	0.4319	0.9839	0.4010
600887	0.6266	0.7532	0.5848	0.3907	0.8335	0.3368	0.7821	0.3316
600919	0.9025	0.3286	0.7401	0.8143	0.8556	0.8143	0.9711	0.0571
600958	0.8315	0.3650	0.5879	0.1679	0.8791	0.4672	0.9158	0.4307
600999	0.6575	0.4344	0.5411	0.5758	0.8522	0.5373	0.5244	0.5013
601006	0.5630	0.6838	0.5758	0.6427	0.8368	0.4730	0.5784	0.7429
601088	0.5784	0.5476	0.5951	0.5064	0.8535	0.4319	0.5803	0.5733
601166	0.5964	0.5064	0.5765	0.5964	0.8329	0.4936	0.8805	0.5064
601169	0.5964	0.4422	0.5572	0.4422	0.8535	0.5681	0.5514	0.4422
601186	0.6780	0.5501	0.5424	0.6298	0.8445	0.6272	0.5116	0.5013
601211	0.7160	0.6613	0.6856	0.6048	0.8296	0.5968	0.9189	0.7016
601229	0.9820	0.3393	0.8198	0.3393	0.8198	0.6429	0.9955	0.6964
601288	0.5857	0.7418	0.5520	0.4560	0.8424	0.5027	0.5623	0.6264
601318	0.6183	0.5604	0.5623	0.3085	0.8329	0.4447	0.8882	0.4781
601328	0.5585	0.4961	0.5360	0.5501	0.8509	0.6632	1.0000	0.4679
601336	0.6105	0.6667	0.5162	0.6769	0.8631	0.6429	0.5519	0.5136
601390	0.6838	0.5527	0.5553	0.5681	0.8683	0.5398	0.5219	0.5604
601398	0.5983	0.6864	0.5514	0.4010	0.8406	0.4653	0.8965	0.5090
601601	0.6420	0.6041	0.5784	0.4319	0.8175	0.4679	0.6247	0.5398
601628	0.6587	0.4627	0.6215	0.4627	0.8560	0.3959	0.6433	0.4679
601668	0.5765	0.5347	0.5386	0.4884	0.8515	0.6041	0.9942	0.5090
601669	0.6000	0.4389	0.5438	0.4785	0.8678	0.5941	0.6050	0.4917
601688	0.6518	0.4334	0.5223	0.5352	0.8606	0.5405	0.6695	0.4204
601766	0.6285	0.4293	0.5521	0.5270	0.8663	0.4807	0.8091	0.4653
601800	0.6037	0.3627	0.5975	0.6408	0.8491	0.6092	0.6267	0.3592
601818	0.6011	0.4763	0.5907	0.3538	0.8459	0.5432	0.5969	0.3259
601857	0.6530	0.3907	0.5488	0.6298	0.8168	0.4781	0.6594	0.3393
601878	0.9808	0.7407	0.8365	0.8519	0.8077	1.0000	1.0000	1.0000
601881	0.8098	0.5532	0.8804	0.1915	0.8641	0.4894	0.8478	0.4681
601985	0.8227	0.7937	0.7032	0.3810	0.9084	0.3254	0.6355	0.1905
601988	0.5585	0.3959	0.5000	0.5630	0.8265	0.6530	0.9582	0.6607
601989	0.6407	0.3445	0.5938	0.2776	0.8348	0.3650	0.6594	0.3959
603993	0.6098	0.3711	0.5941	0.4180	0.8765	0.4102	0.5108	0.4102
Avg.	0.6434	0.5388	0.5911	0.4863	0.8429	0.5039	0.6976	0.5137