

Predicting Stock Trends through Technical Analysis and Nearest Neighbor Classification

Lamartine Almeida Teixeira
Department of Computing Systems
University of Pernambuco
Recife, Brazil
lat@dsc.upe.br

Adriano Lorena Inácio de Oliveira
Informatics Center
Federal University of Pernambuco
Recife, Brazil
alio@cin.ufpe.br

Abstract—This paper presents the results of method designed to predict price trends in the stock market. Our first and foremost objective is to study the feasibility of the practical use of an intelligent prediction system exclusively based on the history of daily stock closing prices and volumes. To this end we propose a technique that consists of a combination of a nearest neighbor classifier and some well known tools of technical analysis, namely, stop loss, stop gain and RSI filter. For assessing the potential use of the proposed method in practice we compared the results obtained to the results that would be obtained by adopting a buy-and-hold strategy. The key performance measure in this comparison was profitability. The proposed method was shown to generate considerable higher profits than buy-and-hold for most of the companies, with few buy operations generated and, consequently, minimizing the risk of market exposure.

Keywords—stock trend prediction, financial forecasting, machine learning, nearest neighbor prediction

I. INTRODUCTION

Prediction of price movements in the stock market is generally believed to be a very difficult task. A well known hypothesis amongst academics, the Efficient Market Hypothesis [1], suggests that prices immediately reflect all the available information and the only thing that causes security prices to change is new information. Therefore, as the arrival of new information is unpredictable, prices in the market appear to be randomly generated. Consequently, it would not be possible to consistently earn excess returns using any available information. According to this hypothesis, the best choice in the market would be to follow the buy-and-hold strategy, which consists of not trying to predict the price movements, i.e. buying and never sell the securities.

Several researches debate about the credibility of the Efficient Market Hypothesis. We can cite the study performed by Haugen [2] and Los [3]. Haugen presents a revision about the deficiencies of this hypothesis, and Los performed experiments considering the six main Asian markets. Los stated that none of these markets presented the efficient behavior suggested by that hypothesis. Besides these works, the practical experience in trading suggests the existence of price trends and that attempts to predict future price changes can generate good returns.

In practice, two approaches are commonly used to analyze price movements, and consequently to predict price trends of

securities. The first is the Fundamental Analysis, which utilizes economic factors to estimate the intrinsic values of the securities. The second approach, known as Technical Analysis, is based on the principles of the Dow Theory [4] and uses the history of prices to predict future movements. The approach used in technical analysis can be formulated as a pattern recognition problem, where the inputs are derived from the history of prices and the output is an estimate of the price or an estimate of the prices trend. Several alternatives to approach this type of problem have been proposed, which range from traditional statistical modeling to methods based on computational intelligence.

Since the 1980's many attempts have been made to predict the stock markets. Vanstone and Tan [8] surveyed the works in this area and classified them in the following topics: time series ([9], [10]), pattern recognition and classification ([11], [12], [13], [14], [15], [21]), optimization ([16], [17]) and hybrid methods ([6], [18], [19], [20]). The first work we found in this area was published in 1988 [7] and used feedforward neural networks to detect regularities in prices time series. Another interesting work, which we consider our main reference for this paper, was the recent research performed by Kwon and Moon [6], which uses a hybrid approach based on the context of the test day. The main goal of that work was to predict future price changes by using technical indicators as inputs. This prediction was based on regression with recurrent neural networks, whose weights were optimized by a combination of genetic algorithms and the traditional backpropagation. The method was tested with 36 stocks, considering a period of 13 years, and was capable of generating considerable higher profits than the buy-and-hold strategy.

The researches in this area are highly diverse regarding the approaches used to model the problem and the metrics used to evaluate the techniques. However, for most of these works it is not clear if the techniques proposed would be capable of generating profits in practice. Therefore our general purpose in this paper is to present a price trend prediction technique and evaluate it according to its capability to generate profits. In this paper, our focus is predicting price trends in the stock market by using some common tools of technical analysis and the well known k-NN algorithm [5]. It is important to emphasize that these common tools of technical analysis, like technical indicators, stop loss, stop gain and RSI filters, are very used by traders in practice, but they are usually ignored in academic research. What we propose is a method that puts together a well

known classifier and some tools that are commonly used in practical trading, with the purpose of investigating the feasibility of using an intelligent trading system exclusively based on data derived from security prices and volume of trades. To evaluate the capability of the proposed system in generating profits we compare its results to the profits that would be obtained by using a buy-and-hold strategy, which is the best alternative according to the Efficient Market Hypothesis.

There are a number of classifiers that are known to achieve better results in terms of generalization than k-NN for many real problems. However, in this first moment we wanted a simple and fast algorithm, since our purpose now is to analyze the influence in the results of some technical analysis tools, like the stops and the RSI filter.

The paper is organized as follows. The basic concepts of technical analysis are presented in section II, while section III presents the proposed method, including a brief revision of the k-NN algorithm, the data sets and the input features, as well as the trading model considered. In section IV we perform some experiments in order to assess the economic significance of the method. In section V we provide some concluding remarks.

II. BASIC CONCEPTS OF TECHNICAL ANALYSIS

According to Murphy [4], technical analysis is the study of market action, primarily using charts, aiming to forecast future price trends. The technician observes the market action via two main variables, the prices and the volume of trades. The most important premise of this type of analysis is that the market action discounts everything. It means the technician believes that anything that can possibly affect the market is already reflected in the prices, as well as that all the new information will be immediately reflected in those prices. As a consequence, all the technician needs is to analyze the history of prices.

The main tools of the technical analysis are the volume and price charts. Based on the information of prices and volume the technical indicators [4] are built. Technical indicators are mathematical formulas that are applied to the price or volume data of a security for modeling some aspect of the movement of those prices. Some indicators, for example, use only closing prices, while others incorporate the volume of trades into their formulas. The most used indicators are those based on moving averages and the oscillators. Those based on moving averages are used to smooth the price movements and, consequently, to make the identification of trends easier. On the other hand, the oscillators are indicators usually employed for identifying momentum.

Initially we have selected four indicators to use in this work. Moving Averages, Relative Strength Index (RSI) and Stochastics are also considered in the work in [6], while the Bollinger Bands are not considered. There is a high diversity of technical indicators in the market. The choice of the technical indicators used in our work was made considering: (i) selecting some of the most widely used by traders in practice and (ii) selecting indicators of different categories, some based on moving averages and some oscillators. Thus it would be interesting to analyze their relevance in practice.

A. Moving Averages

We used in this work the Simple Moving Average (SMA), which is formed by computing the average price of a security over a specified number of periods. The calculation is repeated for each day on the chart and then joined to form a smooth curve, which makes easier to identify the direction of the trend. The SMA formula for a period of n is presented in (1).

$$SMA(t) = \frac{1}{n} \sum_{i=t-n}^t x(i) \quad (1)$$

B. Relative Strength Index (RSI)

The RSI is one of the most popular momentum oscillators. It compares the magnitude of the recent gains to the magnitude of the recent losses and generates a number that ranges from 0 to 100. The technicians say that generally when the RSI rises above 30 it is considered a bullish signal, i.e. it is considered a signal that prices tend to rise. Conversely, when it falls below 70 it is considered a bearish signal, i.e. a signal that prices are likely to fall. The RSI is computed using (2), (3) and (4).

$$RSI = 100 - \frac{100}{1 + RS}, \text{ where } RS = \frac{avgGain}{avgLost} \quad (2)$$

$$avgGain = (\text{total of gains during past } n \text{ periods}) \div n \quad (3)$$

$$avgLost = (\text{total of losses during past } n \text{ periods}) \div n \quad (4)$$

C. Stochastics

The stochastics oscillator compare the location of the current close price of a security relative to its price range over a certain number of periods. This indicator is composed by two lines, the fast (%K) and the slow (%D), which are calculated according to (5) and (6). According to Murphy [4], closing levels that are consistently near the top of the range indicate accumulation (buying pressure) and those near the bottom of the range indicate distribution (selling pressure).

$$\%K = 100 \times \frac{recentClose - lowestLow(n)}{highestHigh(n) - lowestLow(n)} \quad (5)$$

$$\%D = 3\text{-period moving average of \%K} \quad (6)$$

D. Bollinger Bands

The Bollinger Bands were designed to compare the volatility and relative price levels over a period time. It consists of three bands that encompass the majority of the price action. The bands are: (1) an SMA in the middle; (2) an upper band, calculated as an SMA plus a number of standard deviations; (3) a lower band, calculated as an SMA minus a number of standard deviations. In addition to identify relative price levels and volatility, Bollinger Bands can also be combined with price action to generate signals and foreshadow significant moves.

III. USING K-NN TO STOCK TREND PREDICTION

A. K-NN

The nearest neighbor (NN) classifier is one of the simplest machine learning algorithms. In this type of classifier, a new pattern is classified according to its similarity with the available

training patterns. To measure this similarity, the Euclidean distance between the new pattern being classified and all the other patterns in training set is computed. Next, the class of the training pattern with the smaller distance is assigned to the new pattern. The k-nearest neighbor classifier (k-NN) is similar to the NN. In the k-NN algorithm a class is assigned according to the most common class amongst its k nearest neighbors.

k-NN training is extremely fast, since it consists solely of storing all training patterns. This is a great advantage for our proposed method, since we intend to analyze a high number of stocks each day. On the other hand, it is known that for many problems, techniques like MLP neural networks present better results than k-NN in terms of classification accuracy. However depending on the problem, these techniques sometimes present a prohibitive performance. Specifically in the task of stock market forecasting, performance is a critical issue, since it would be desirable to train the classifier with many stocks and possibly retrain the classifier every day.

B. Inputs

Fifteen real stocks from Sao Paulo Stock Exchange (Bovespa) [25] were used for testing the proposed method and comparing it with the buy-and-hold approach. Our goal was to consider in the experiments a time period long enough to capture a high diversity in price movements. Therefore, we decided to cover the period from April 1, 1998 to March 9, 2009.

We extracted some features from the history of prices to use as inputs to k-NN for each day t . Some of these inputs were also used in the work of Kwon and Moon [6]. The inputs are reported in Table I, where $x(t)$, $x_h(t)$ and $x_l(t)$ means, respectively, the stock's close, high and low price on day t , $v(t)$ is the volume of trades on day t , $bollinger_{upper}$ and $bollinger_{lower}$ are the upper and lower Bollinger bands, PMA_s and PMA_l are the short-term and long-term simple moving averages applied to prices, and VMA_s and VMA_l are the short-term and long-term simple moving averages applied to volume.

TABLE I. FEATURES USED AS INPUTS TO K-NN

$I_1 = RSI(t)$
$I_2 = (x(t) - bollinger_{upper}) / bollinger_{upper}$
$I_3 = (x(t) - bollinger_{lower}) / bollinger_{lower}$
$I_4 = \%K(t)$
$I_5 = \%D(t)$
$I_6 = \%K(t) - \%K(t-1)$
$I_7 = \%D(t) - \%D(t-1)$
$I_8 = (x(t) - x(t-1)) / x(t-1)$
$I_9 = (x(t) - x_l(t)) / (x_h(t) - x_l(t))$
$I_{10} = (PMA_s(t) - PMA_s(t-1)) / PMA_s(t-1)$
$I_{11} = (PMA_l(t) - PMA_l(t-1)) / PMA_l(t-1)$
$I_{12} = (PMA_s(t) - PMA_l(t-1)) / PMA_l(t-1)$
$I_{13} = (x(t) - PMA_l(t)) / PMA_l(t)$

$$I_{14} = (x(t) - \min(x(t), \dots, x(t-5)) / \min(x(t), \dots, x(t-5)))$$

$$I_{15} = (x(t) - \max(x(t), \dots, x(t-5)) / \max(x(t), \dots, x(t-5)))$$

$$I_{16} = (v(t) - v(t-1)) / v(t-1)$$

$$I_{17} = (VMA_s(t) - VMA_s(t-1)) / VMA_s(t-1)$$

$$I_{18} = (VMA_l(t) - VMA_l(t-1)) / VMA_l(t-1)$$

$$I_{19} = (VMA_s(t) - VMA_l(t-1)) / VMA_l(t-1)$$

$$I_{20} = (v(t) - VMA_l(t)) / VMA_l(t)$$

$$I_{21} = (v(t) - \min(v(t), \dots, v(t-5)) / \min(v(t), \dots, v(t-5)))$$

$$I_{22} = (v(t) - \max(v(t), \dots, v(t-5)) / \max(v(t), \dots, v(t-5)))$$

The whole data set was divided into 10 subsets, where each subset contains about one year of trades. Next, we use a walk-forward testing model, where each time we select three subsequent subsets for training and the next subset for testing. Thus we have selected seven subsets to test, each one corresponding to approximately one year of price movements. We separate the data sets according to Fig 1. A similar procedure was also employed by Kwon and Moon [6].

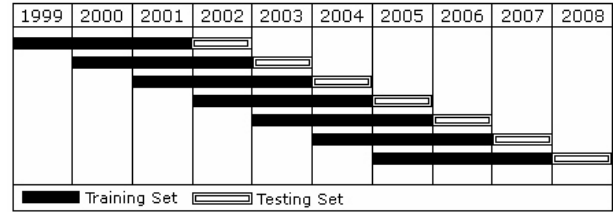


Figure 1. Scheme of data sets division for testing

C. Trading Model

In this paper, we considered a daily trading model. We defined to our experiments an initial cash balance and for each test day t the classifier's output is used to start a buy (long) or sell (short) operation. The operations are performed by using all the available cash balance and respecting the minimum number of 100 shares per operation, which is a rule in Brazilian stock market. If the classifier's output for day t is *buy*, then a long position is performed considering the stock's closing price and all the available cash balance. If the classifier's output is *sell* and there are any stock available in custody, then all the stocks are sold. Otherwise, no operation is performed if the generated signal is *keep*. It is important to stress that we consider in our model the real world transaction costs of the Brazilian market.

We are also considering in our trading model some trader's common tools, the *stops*. The stop loss was used to limit the loss in a single trade to 3%. The stop gain was used to protect the gain when it approaches to 10%. Therefore, if after a buy operation the prices fall under the 3% limit or rise more than 10%, then all the available stocks in custody are sold. In addition, we also investigate the use of a RSI filter, which aims to avoid long positions when the stock is too overbought and, consequently, to reduce the risk of that trade. Thus, whatever the classifier output, when using this type of filter, the buy operation is not performed if the RSI readings are higher than 70.

D. Performance Criteria

There are a number of computational intelligence techniques applied to financial forecasting, however it is not clear for most of these works if the proposed techniques would be capable of generating profits in practice. Therefore, the main parameter adopted in our experiments for evaluating the performance is the profit obtained during the test period. We define a trading model, calculate the accumulated profit in the test period and then compare to the buy-and-hold strategy. We also evaluate the results in terms of the percentage of correctly classified instances.

IV. EXPERIMENTS

A. Experimental Setup

To begin each experiment we consider an initial cash balance of R\$ 100,000.00 (Brazilian real). For the transaction cost, we consider a fixed value of R\$ 5.00, which is a real charged value by some Brazilian stockbrokers.

Before performing the experiments we also had to choose some parameters for constructing the technical indicators. All of these parameters were set according to the values commonly used by traders in practice and also cited by Murphy [4]. The short-term and long-term simple moving average periods were set, respectively, to the values 10 and 21. The RSI period was 14. The fast and slow stochastics were set to the values 14 and 3. To build the Bollinger bands, we used a simple moving average of eight days and two standard deviations for upper and lower bands.

B. Experimental Results

The results of the first set of experiments are presented in Fig. 2, which depicts the accumulated profit during the seven years of testing. This figure compares the profit obtained by proposed method to the profit that would be obtained by the buy-and-hold strategy. It shows the results obtained by using the k-NN combined with stop loss, stop gain and RSI filter. We see that the profit of the classifier was lower than buy-and-hold only in 3 out of 15 stocks (BBDC4, CRUZ3 and ELET6), similar profits were achieved for CMIG4, and the classifier

outperforms buy-and-hold for the other 12 stocks.

We see that the applicability of the stop loss, stop gain and RSI filter must be individually analyzed for each stock, since there were increments of more than 50% in the accumulated profits for some stocks, while for others the use of these strategies reduced the profits. For stocks CSNA3, ITAU4, PETR4 and VALE5, for example, the use of some combination of stops and RSI filter increased the profits in 45%, 42%, 51% and 34%, respectively. On the other hand, ITSA4 and USIM5 had the profitability reduced in 25% and 22%, respectively.

The detailed results for the best configuration found are presented in Table II. We mean configuration by combinations of the k-NN and the trading strategies of stop loss, stop gain and RSI filter (results shown in Fig. 2). This table shows the average classifier precision per testing year, which indicates the percentage of correctly generated trading signals, the difference between the obtained profits and the buy-and-hold profits, and the average number of buy operations per testing year. With regard to the average number of buy operations per year of test, we obtained very uniform values, about 40 operations a year, or the equivalent to less than 4 operations a month. We believe this small number of operations is a desired feature for an automatic trading system, since the risk of capital exposure to the market would be minimized.

In stock market predictions we see that a low classifier precision not necessarily means a poor classifier in terms of capability of generating profits. As presented in Table II, even with the low precision of the classifier, good results in terms of profitability were achieved. Analyzing the results, we see that the classifier outperformed the buy-and-hold strategy in 12 out of 15 stocks. For 8 of these 12 stocks we obtained 30% higher profits or more. For the stock ITAU4, for example, we had a precision of 37.4%, however we obtained a profit 115.99% higher than buy-and-hold.

Based on the low percentage of correctly classified instances, we decided to perform an experiment to measure the profit that would be gained if it would be possible to achieve 50% precision with our classifier. Therefore, to perform the experiments we artificially improved the classifier precision, so

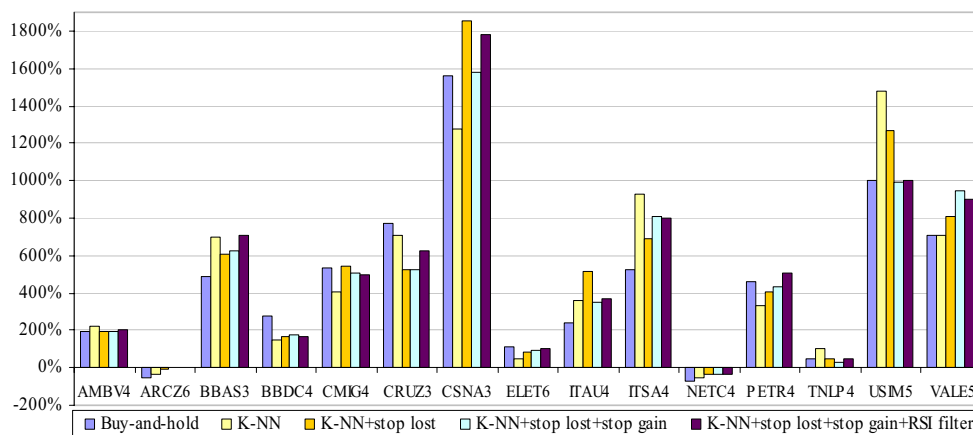


Figure 2. Results compared to buy-and-hold estategy

that it could have a precision very close to 50%. Then we proceeded to perform the tests and the results are presented in Fig. 3.

TABLE II. AVERAGE RESULTS FOR THE BEST CONFIGURATION

Stock	Classifier Precision	Difference to buy-and-hold	Buy operations per year
AMBV4	40,78%	+12,20%	27,85
ARCZ6	34,47%	+108,43%	44,57
BBAS3	36,12%	+45,34%	47,28
BBDC4	36,81%	-36,78%	44,14
CMIG4	36,99%	+1,10%	48,28
CRUZ3	36,86%	-8,09%	38,57
CSNA3	39,16%	+18,86%	48,28
ELET6	37,34%	-10,90%	54,28
ITAU4	37,40%	+115,99%	44,85
ITSA4	37,20%	+76,83%	37,14
NETC4	37,87%	+51,98%	52,00
PETR4	34,59%	+8,13%	42,14
TNLP4	35,96%	+95,20%	37,71
USIM5	40,00%	+47,18%	43,28
VALE5	36,55%	+33,12%	40,14

The original classifier had an average precision of approximately 40%. Therefore, in the new experiment the classifier precision was artificially incremented by about 10%. The results of Fig. 3 show that this increment in the classifier precision of only 10% generated a high increment in the profits gained for all the stocks. In addition, we also see that the lower profits that would be obtained with this hypothetical classifier would be for stock AMBV4, which presented a profit 5 times higher than the profit obtained by following the buy-and-hold strategy. Considering the case of NETC4, for example, the accumulated profit would be 80 times higher than that achieved by the buy-and-hold strategy. These results suggest that our classifier precision is suitable for this type of problem and that small future improvements in the precision of the classifier can make a great advance in the profits, that is, in the feasibility of using this type of trading system in practice.

Some improvements in the classifier precision could be obtained with the use of other classifiers, like multilayer perceptron neural networks (MLP) [26], MLP-PSO [22], Extreme Learning Machines (ELM) [23] or Support Vector Machines (SVM) [24], which presented better generalization ability than k-NN for several real problems. It is known, for instance, that for many problems the MLP achieves better results than k-NN in terms of classification accuracy. However the traditional MLP training algorithms, like backpropagation, as well as the more recent, like the PSO-based, in general are very slow, which could make prohibitive the practical use in stock market forecasting, since in such systems it would be desirable to retrain the classifier in every trading day. On the other hand, the recently proposed ELM classifier may be an

option for use in such system, since it is much faster to train than traditional MLP or SVMs, but yields similar or even better classification accuracy for many real problems [23]. Therefore ELM certainly will be considered in our future works.

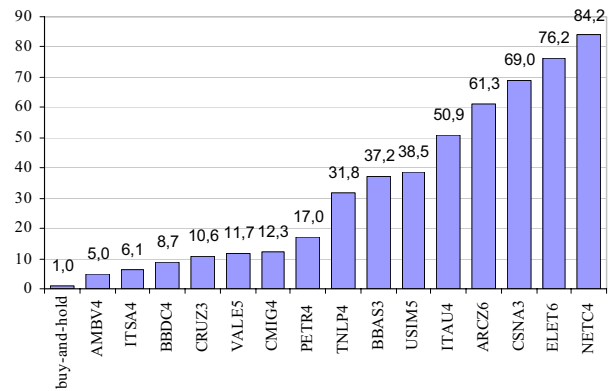


Figure 3. Results for a hypothetical classifier versus buy-and-hold strategy

Table III presents the profits that would be obtained by creating a portfolio composed of all the stocks considered in this work, and following the trading signals generated by the method proposed in this paper. In this table we present the results to the four strategies experimented: (i) k-NN, (ii) k-NN combined with stop loss, (iii) k-NN combined with stop loss and stop gain, and (iv) k-NN combined with stop loss, stop gain and RSI filter. We see that in all the cases our trading system outperformed the buy-and-hold strategy. According to those results, the best returns would be obtained by using the k-NN combined with stop loss, and k-NN combined with stop loss, stop gain and RSI filter. We see that applying this trading system to construct stock's portfolios would be interesting, since the trader would be able to compensate losses for some stocks with gains in other stocks.

TABLE III. PROFITS OF A HYPOTHETIC PORTFOLIO

Strategy	Accumulated Profit	Difference to buy-and-hold
k-NN	487,74%	34,50%
k-NN+stop loss	511,01%	57,76%
k-NN+stop loss+stop gain	481,95%	28,70%
k-NN+stop loss+stop gain+RSI filter	511,35%	58,10%

V. CONCLUSIONS

In this paper a stock trading system is proposed, which is exclusively based on the history of the daily closing prices and trading volumes. We used the well known k-NN classifier and investigated the feasibility of using it in real market conditions, considering real companies of São Paulo Stock Exchange (Bovespa) and transaction costs. In addition, we proposed the use of some common tools of technical analysis, like technical indicators, stop loss, stop gain and RSI filters, which are commonly used in practice by trades, but are usually ignored in academic research. Thus we proposed a method that puts

together a well known classifier and some tools that are really used in practice, with the purpose of investigating the feasibility of using an intelligent trading system in real market conditions. It was shown that our method can achieve good results in terms of profitability. Comparing the results with the profits that would be obtained with the buy-and-hold strategy, we see that our method performs better for 12 of 15 stocks considered in the experiments. Therefore, we believe that using this approach for predicting real short term stock trends is possible.

As future research, we will investigate the use of other classifiers, like Support Vector Machines (SVMs) [24], Extreme Learning Machines (ELMs) [23] and MLP-PSO [22], which were shown to achieve a higher generalization potential than k-NN for several real problems. In addition we will investigate the possibility of including some Asian market indices as input features, since the Asian markets close before the Brazilian market open.

REFERENCES

- [1] E. F. Fama, Efficient Capital Markets: A Review of Theory and Empirical Work. *Journal of Finance*, Volume 25, Issue 2, Papers and Proceedings of the Twenty-Eighth Annual Meeting of the American Finance Association, New York, N.Y. pp. 383-417, May 1970.
- [2] R. A. Haugen, *The New Finance: The Case Against Efficient Markets*, Prentice Hall, 1999.
- [3] C. A. Los, Nonparametric efficiency testing of Asian markets using weekly data. *Advances in Econometrics*, volume 14, pp. 329-363, 2000.
- [4] J. J. Murphy, *Technical Analysis of the Financial Markets*, New York Institute of Finance, 1999.
- [5] T. M. Cover and P. E. Hart, Nearest neighbor pattern classification, *IEEE Trans. Inform. Theory*, vol. 13, pp. 21-27, 1967.
- [6] Y. Kwon and B. Moon, A Hybrid Neurogenetic Approach for Stock Forecasting, *IEEE Transactions on Neural Networks*, Vol. 18, No. 3, pp. 851-864, 2007.
- [7] H. White, Economic prediction using neural networks: a case of IBM daily stock returns, *IEEE Int. Conference on Neural Networks*, vol. 2, pp. 451-458, 1988.
- [8] B. Vanstone and C. Tan. A survey of the application of soft computing to investment and financial trading. *Proceedings of the Australian and New Zealand Intelligent Information Systems Conference*, p. 211-216, Sydney, 2003.
- [9] L. J. Cao, F. E. H. Tay, Support Vector Machine With Adaptive Parameters in Financial Time Series Forecasting, *IEEE Transactions on Neural Networks*, Vol. 14, no. 6, 1506-1518, 2003.
- [10] V. Nagarajan, Y. Wu, M. Liu, Q. Wang, Forecast Studies for Financial Markets using Technical Analysis, *International Conference on Control and Automation (ICCA)*, 259-264, 2005.
- [11] E. W. Saad, D. V. Prokhorov, D. C. Wunsch; Comparative Study of Stock Trend Prediction Using Time Delay, Recurrent and Probabilistic Neural Networks; *IEEE Transactions on Neural Networks*, Vol. 9, No. 6, 1456-1470, 1998.
- [12] L. Nanni, Multi-resolution subspace for financial trading, *Pattern Recognition Letters* 27, 109-115, 2006.
- [13] Y. Sai, Z. Yuan, Mining Stock Market Tendency by RS-Based Support Vector Machines, *IEEE International Conference on Granular Computing*, p. 659-664, 2007.
- [14] W. Leigh, C. J. Frohlich, S. Hornik, R. L. Purvis, T. L. Roberts, Trading With a Stock Chart Heuristic, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 38, No. 1, 93-104, 2008.
- [15] X. Guo, X. Liang, X. Li, A Stock Pattern Recognition Algorithm Based on Neural Networks, *Third International Conference on Natural Computation (ICNC)*, 518-522, 2007.
- [16] J. Chang, S. Hsu, The Construction of Stock's Portfolios by Using Particle Swarm Optimization, *Proceedings of the Second International Conference on Innovative Computing, Information and Control*, p. 390, 2007.
- [17] P. Tan. Using Genetic Algorithm to optimize an oscillator-based market timing system, *Proceedings of the Second Singapore International Conference on Intelligent Systems (SPICIS)*, 1994.
- [18] M. O. Afolabi, O. Olude, Predicting Stock Prices Using a Hybrid Kohonen Self Organizing Map (SOM), *Proceedings of the 40th Hawaii International Conference on System Sciences*, p. 48, 2007.
- [19] J. Mandziuk, M. Jaruszewicz, Neuro-evolutionary approach to stock market prediction, *Proceedings of International Joint Conference on Neural Networks*, Orlando, Florida, USA, 2515-2520, 2007.
- [20] H. Kim, K. Shin, A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets, *Applied Soft Computing*, 7, 569-576, 2007.
- [21] D. Bao, Z. Yang, Intelligent stock trading system by turning point confirming and probabilistic reasoning, *Expert Systems with Applications* 34, p. 620-627, 2008.
- [22] L. A. Teixeira, F. T. Oliveira, A. L. Oliveira and C. J. Filho, Adjusting Weights and Architecture of Neural Networks through PSO with Time-Varying Parameters and Early Stopping. In *Proceedings of the 2008 10th Brazilian Symposium on Neural Networks - SBRN*, pp. 33-38, IEEE Computer Society Press.
- [23] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, Extreme Learning Machine: Theory and Applications, *Neurocomputing*, vol. 70, pp. 489-501, 2006.
- [24] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning*, v. 20, 273-297, 1995.
- [25] *Brazilian Mercantile & Futures Exchange and São Paulo Stock Exchange (BM&FBovespa)*, <http://www.bovespa.com.br/indexi.asp>
- [26] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, Massachusetts, MIT Press, 1995.