mathematics

MDPI

Article

A Nonlinear Technical Indicator Selection Approach for Stock Markets. Application to the Chinese Stock Market

Gerardo Alfonso [1,2,*] and Daniel R. Ramirez [2]

[1] Shenwan & Hongyuan Securities Co., Ltd. (SWS), Shanghai 200031, China
[2] Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, 41092 Sevilla, Spain; dani@us.es
[*] Correspondence: ga284@cantab.net

Received: 1 July 2020; Accepted: 5 August 2020; Published: 6 August 2020

Abstract: In this paper we present a combinatorial nonlinear technical indicator approach for the identification of appropriate combinations of stock technical indicators as inputs in non-linear models. This approach is illustrated with the example of Chinese stock indexes and 35 different stock technical indicators using neural networks as the chosen non-linear method. Stock market technical indicators can generate contradictory signals regarding the future performance of the stock analyzed. Furthermore, some non-linear methods, such as neural networks, can have poor generalization power when dealing with problems of high dimensionality due to the issue of local minima. Therefore, non-linear approaches that can identify appropriate combinations of input variables are of clear importance. It will be shown that the proposed approach, when using neural networks as classifiers, generates error rates lower than those using directly neural networks without dimensionality reduction. It will also be shown that merely increasing the number of neurons does not increase the accuracy. The approach proposed in this article is illustrated with an application to the stock market using neural networks but it could be applied to other fields and it can also be used with other non-linear techniques such as for instance support vector machines.

Keywords: feature selection; neural networks; stocks

1. Introduction

Over the last few decades there has been an increase in non-linear forecasting techniques, such as for instance neural networks (NN) and support vector machines (SVM). The non-linearity of the stock market performance has been mentioned by many researchers such as for instance Vrbka and Rowland [1]. In fact, the non-linearity of the stock market was one of the reasons behind choosing neural networks (a non-linear approach) by Vrbka and Rowland [1] as a model to forecast stocks prices in Prague Stock Exchange. In this paper the authors successfully applied multilayer perceptron and radial basis networks to that particular stock market. While neural networks are an important stock forecasting technique it should be noted that, as in any other technique, it has limitations. Horack and Krulicky [2] did in this regard an interesting comparison between the exponential time series alignment method and the time series alignment with neural networks method. The authors highlighted the importance of neural networks in the field of stock forecasting mentioning that generally neural networks provide better forecast than traditional methods. However, they also concluded that in their example, applied to a volatile stock, the traditional forecasting method generated better results than the neural network. This highlights the importance of using the appropriate stock forecasting techniques with papers in the existing literature finding less than optimal results for some popular techniques. For instance, Groda and Vrbka [3] concluded that the Box–Jenkins method is not an appropriate method in the case of stocks listed in the Prague Stock Exchange.

An important development in recent years is the very large increase in data available in many disciplines. It is relatively frequent to have a high number of variables that can potentially have an impact in non-linear processes, see Guyon and Elisseeff [4]. There is substantial research covering the topic of variable selection using linear methods, such as for instance Hocking [5], but these techniques might not be ideal when intended for non-linear modelling. There is relatively little literature covering the issue of variable selection of non-linear processes from a combinatorial approach. The basic assumption is that in non-linear processes the way in which different independent variables interact with each other can be highly complex. Identifying which combination of variables work better for a non-linear problem is clearly not trivial, see Yuan and Lin [6]. There are some interesting methods, such as for instance Rech and Terasvirt [7], using polynomial approximation. The drawback of this type of approach is that it is only applicable when there is a relatively small number of variables.

Ye and Sun [8] proposed an iterative method in which starting from all the variables considered one, of the variables is dropped and the resulting set of variables is used, using neural networks, and then the results compared.

The stock market, as many other fields, has seen a large increase in the number of data available. More specifically, many researchers and practitioners have developed large amount of technical indicators. Getting into the details of these indicators is beyond the scope of this article but it is important to mention that they are typically constructed using historical data such as for instance the closing price of a stock. A moving average is a well-known technical indicator [9–15]. A simple moving average can be constructed as the average of the closing price of a certain stock or index over a certain period of time. There are multiple different technical indicators and strategies based on those indicators with diverse levels of profitability [16–19]. Neural networks have been applied successfully to the field of stock market forecasting [20–25]. One of the frequently mentioned drawbacks of neural networks is the issue of local minima [26–29] which can cause the neural network to generalize poorly or in other words, generate poor forecast when faced with new data. In this regard there has been a focus on reducing the dimensionality of the data to avoid having the neural network stuck in this local minima [30–34]. In this paper we propose a non-linear combinatorial approach for variable selection applied to stock market technical indicators. Given the number of possible technical indicators that are available today, is clear that not all possible combinations can be tested.

In this work we use a pool of 35 technical indicators, so the number of possible combinations is staggering. Because of that the proposed approach resorts to randomization. The algorithm starts by generating a preset number of combinations each with a size of half the number of available indicators. The reason for using such size is because the maximum number of combinations for a given n and k, that is the binomial coefficient n choose k, is attained for $k = \lfloor \frac{n}{2} \rfloor$. This is evident by inspecting the Pascal's Triangle, but can be proved using the properties of the Newton binomial. From this initial set of combinations, new combinations are generated randomly retaining the best ones in an iterative process that ends when the stop condition is met. The proposed method is shown to improve the baseline approach, that is using all the available indicators, when applied to the task of predicting market trends. As a case study for the proposed method, this paper focuses on the chinese stock market. China stock market is a very dynamic and of increasingly importance due to the economic grow of China. The most relevant chinese stock indexes have been considered. The strategy described in this paper has obtained good results and better combinations of technical indicators have been identified for such indexes.

The remaining of the paper is organized as follows: Section 2 presents the proposed approach. Section 3 shows the application of the proposed approach to the China stock market. The results are discussed in Sections 4 and 5 presents the conclusions of the paper.

2. Technical Indicator Selection Approach

Let $X_T^i(t)$ be tuple of T values of the i-th technical indicator from a pool of up to N nonlinear technical indicators computed at the time period t (usually the time period is measured in days, but could be weeks, months or whatever), such as for instance a moving average:

$$X_T^i(t) = \{X^i(t-(T-1)), X^i(t-(T-2)), \ldots, X^i(t)\} \tag{1}$$

where $X^i(t)$ is the value of the i-th technical indicator evaluated at time t.

Let $R_T(t)$ be a vector that groups the direction of the change of the price of a stock or index, in the period from $t-(T-1)$ to t, that is $R_T = \{0,1\}^T$ with 0 meaning that the stock increases or remains constant at the end of a period, 1 meaning otherwise. Additionally, let $\hat{\phi}$ define a nonlinear mapping from $X_T^i(t)$ to $R_T(t)$

$$\hat{\phi}(X_T^1(t), X_T^2(t), \ldots, X_T^N(t)) = \hat{R}_T(t) \tag{2}$$

where $\hat{}_T(t)$ is an estimation of $R_T(t)$. In order to guarantee that an estimator $\hat{\phi}$ can be found is necessary to assume the following:

Assumption 1 (Ground truth function existence)**.** *It is assumed that, ϕ, a mapping from $X_T^i(t)$ to $R_T(t)$ exists.*

Technical indicators in the stock market can produce contradictory signals and some non-linear techniques can have local minima issues when using high dimensionality input variables, (large N or T value). Therefore it might be convenient to find a combination of the technical indicators X_T^i rather than using all N available indicators. We present a combinatorial selection approach for technical indicators that do not have to be linear variable combination in non-linear processes.

The steps are as follows:

1. Split the available instances of $X_T^i(t)$ and $R_T(t)$ data into two subsets, an estimation subset $S_e \triangleq \{X_{T,e}^i(t), R_{T,e}(t)\}$ and a validation (To keep the notation as clear as possible we use here the term "validation" with the meaning of test) subset $S_v \triangleq \{X_{T,v}^i(t), R_{T,v}(t)\}$.
2. Generate C_s, a set of M combinations of $\lfloor \frac{N}{2} \rfloor$ random numbers in the range $\{1, 2, \ldots, N\}$ denoted from i_1 to $i_{\lfloor \frac{N}{2} \rfloor}$. Check for repetition within each combination. If there is repetition, leave out the repeated numbers and keep generating random numbers in the aforementioned range until there is no repetition. Similarly, check that there are no repeated combinations and proceed to replace the redundant ones.
3. For each of the M combinations in C_s, denoted as $I \in C_s$, perform the following steps:

   (3a) Compute $\hat{\phi}$, i.e., the estimated non-linear classification mapping, using any technique of choice. In this case, the nonlinear mapping will have as arguments $X_T^i(t), \forall i \in I$. Using neural networks as an example, this step involves training a neural network with the training data set $\{X_{T,e}^i(t), R_{T,e}(t)\}$ with $i \in I$.

   (3b) Evaluate $\hat{\phi}$ over the validation set. Let $\hat{R}_{T,v}$ denote the estimated classification output of $R_{T,v}$.

   (3c) Calculate the error $\xi_T(t)$ of the non-linear classification approach for each instance in the validation set, so that

   $$\xi_T(t) = \begin{cases} 0 & \text{if } \hat{R}_{T,v}(t) = R_{T,v}(t), \\ 1 & \text{Otherwise} \end{cases}$$

   with the resulting total error being:

   $$\xi^{Total} = \frac{\sum_{\forall t \in S_v} \xi_T(t)}{\text{card}(S_v)} \tag{3}$$

where $\text{card}(S_v)$ denotes the cardinality of S_v. This is therefore the total error for the initial random combination of technical indicators chosen in step 2.

(3d) Randomly generate a single value $i_a \in \{1, 2, ..., N\}$ that would denote a technical indicator candidate to be added to the combination chosen in step 2. Check for repetition with the previously generated $\frac{N}{2}$ values. If there is repetition, randomly generate another value i_a. Repeat this step until there is no repetition.

(3e) Randomly generate a single value $i_r \in \{i_1, \ldots, i_{\frac{N}{2}}\}$. This value will be used later to denote a technical indicator to be removed from the combination chosen in step 2.

(3f) Form a new combination of technical indicator indexes I^{up} as $I^{up} \triangleq I \cup \{i_a\}$. In the same fashion, form a new combination I^{down} as $I^{down} \triangleq I - \{i_r\}$. If some of these combinations are already in C_s repeat either step 3d or 3e until a combination different from those of C_s is obtained.

(3g) As in step 3a, compute two new mappings, denoted $\hat{\phi}^{up}$ and $\hat{\phi}^{down}$ using this case as inputs arguments the technical indicators given by the index combinations I^{up} and I^{down} respectively.

(3h) As in step 3c compute the resulting total error of evaluating the two mappings computed in the previous step over the validation set. Denote these total errors as ξ_{up}^{Total} and ξ_{down}^{Total}.

(3i) Given the three previous index combinations I, I^{up}, I^{down} and their resulting total errors ξ^{Total}, ξ_{up}^{Total} and ξ_{down}^{Total} pick the two combinations with the smallest error.

(3j) Substitute the initial combination I by the two combinations picked in the previous step. This will double the number of combinations in set C_s, i.e., C_s will contain $2M$ combinations after this step.

4. Retain the M combinations in C_s with the smallest error and discard the remaining M combinations with greatest error.
5. Find the minimum ξ^{Total} of all the combinations $I \in C_S$.
6. Repeat step 3 starting from 3d until a certain stop condition is met. Here it is proposed to check if a certain objective error is achieved or a maximum number of iterations is met, so that an infinite loop is avoided.

Remark 1. *This strategy can be easily parallelized as the tasks in step 3 can be done independently for each combination, hence can be done in parallel, with one exception. The latter part of step 3f, that is the rejection of repeated combinations, cannot be done in parallel and must be done sequentially in a separate step outside of step 3.*

Remark 2. *The number of combinations in C_s, that is M, and the number of iterations are related in the sense that similar performances can be achieved by using a greater M and fewer iterations or vice versa. Besides the differences in performance due to the randomized nature of the proposed strategy, a practical difference can lead to one choice or the other. With a greater M one can exploit the parallelizable nature of the algorithm, whereas in the case of a high number of iterations that cannot be made.*

To illustrate the proposed approach consider a simple example with just two combinations (i.e., $M = 2$) and one iteration. If for instance, we assume that we have a total of 6 technical indicators to choose from, i.e., $\{X_5^1, X_5^2, X_5^3, X_5^4, X_5^5, X_5^6\}$, where the sub-index 5 means that each of the indicators are evaluated over five periods of time. There is also a related classification vector R_5 identifying up and down movements in the stock at each time t (in this case $R_5 \in \{0, 1\}^5$). First, the algorithm selects randomly 2 combinations of 3 initial technical indicator indexes. For example:

$$C_s = \{I_1, I_2\} = \{\{3, 5, 6\}, \{1, 2, 3\}\}$$

Then the non-linear classification error is estimated for this configuration; let us assume that the obtained value is (with a slight abuse of notation the parameters of each combination will denote in this example the set of the parameters of both combinations):

$$\xi^{Total} = \{0.6, 0.5\}$$

Then the indexes i_a and i_r are computed for each combination:

$$i_a = \{1, 4\} \;\; i_r = \{6, 2\}.$$

The addition of the new technical indicator index is done randomly, ensuring that there is no repetition, i.e., each input variable (technical indicator) is used only once. The index removed is also computed randomly. Then the combinations I^{up} and I^{down} are formed:

$$I^{up} = \{I_1^{up}, I_2^{up}\} = \{\{3, 5, 6, 1\}, \{1, 2, 3, 4\}\}$$
$$I^{down} = \{I_1^{down}, I_2^{down}\} = \{\{3, 5\}, \{1, 3\}\}$$

Note that in every combination generated so far no index is repeated and that there are not repeated combinations. Once the new combinations are formed, their total classifying errors are computed:

$$\xi_{up}^{Total} = \{0.7, 0.3\}$$
$$\xi_{down}^{Total} = \{0.4, 0.2\}$$
$$\xi^{Total} = \{0.6, 0.5\}$$

For every combination in C_s we choose the two combinations (from I, I^{up} and I^{down}) with the smallest error. From the fist combination we choose then, I_1^{down} and I_1. On the other hand, from the second combination we choose I_2^{up} and I_2^{down}. Therefore, the augmented C_s will consists of:

$$C_s = \{I_1^{down}, I_1, I_2^{up}, I_2^{down}\}$$

with their errors:

$$\xi = \{0.4, 0.6, 0.3, 0.2\}$$

We order the combinations using the error as sort criterion:

$$C_s = \{I_2^{down}, I_2^{up}, I_1^{down}, I_1\}$$

$$\xi = \{0.2, 0.3, 0.4, 0.6\}$$

Now we can reduce the numbers of combinations to its original number ($M = 2$) picking the two with the smallest error:

$$C_s = \{I_2^{down}, I_2^{up}\}$$

with errors:

$$\xi = \{0.2, 0.3\}$$

Then at the end of this first iteration the algorithm would pick the combination I_2^{down} as the best technical indicator combination, as it has the lowest error (0.2). Thus, the technical indicators proposed to forecast the direction of the change of price will be $\{X_5^1, X_5^3\}$.

In practice, the number of combinations and iterations will be determined by the computing power available. The process thus will be repeated until a stop criteria is reached (that is, a maximum number of iterations or a target classification error).

3. Chinese Equity Market Application

3.1. Data and Methodology

The Chinese equity market is an increasingly important market propelled by the large economic expansion of the Chinese economy over the last few decades. The Chinese equity market is divided into two major stock exchanges. The Shanghai and the Shenzhen Stock Exchange with several major stock indexes describing the performance of those markets. We used 6 different stock indexes describing the Chinese stock. For completeness purposes, and to exclude some form of regional bias in the results, we also used to other international (non-Chinese) stock indexes (see Table 1).

Table 1. Stock indexes.

Indexes	
A50	FTSE China A50 index composed of the largest chinese companies.
CSI 300	Largest 300 companies listed in the Shenzhen and Shanghai Stock Exchanges
CSI 800	Largest 800 companies listed in the Shenzhen and Shanghai Stock Exchanges
Shanghai Composite	Index composed of all the stocks listed in the Shanghai Stock Exchange
Shenzhen Component	Index composed of the largest 500 companies in the Shenzhen Stock Exchange
SSE 50	Index composed of the largest 50 companies in the Shanghai Stock Exchange
Euro Stoxx 50	Index composed by 50 of the largest companies from 11 Euro zone countries
New York Composite	Index composed of all the stocks listed in the New York Stock Exchange

As described in the previous section the selection approach is used to forecast the direction of the movement of the stock index (up or down) in the next time period rather than the exact end price for that period. Daily closing prices for all the indexes mentioned in Table 1 were collected from the Bloomberg database for the period from 14 February 2007 to 30 March 2020. The returns of those index can be seen in Figure 1. Positive or zero returns will result in an R_T value of 0 and 1 otherwise.

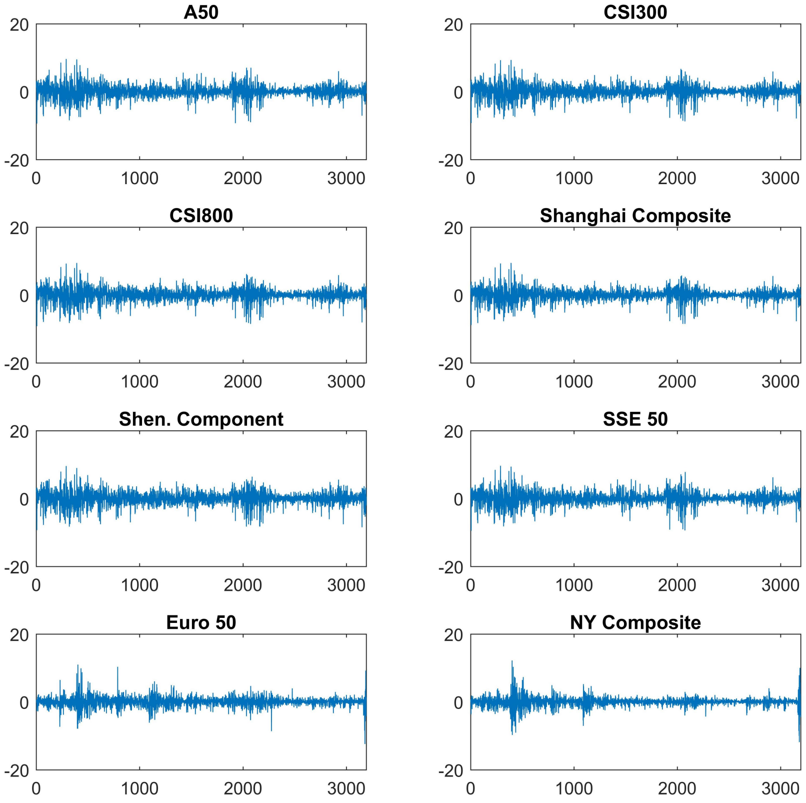

Figure 1. Returns of the indexes (measured in percentages).

The proposed approach was used to identify an appropriate combination of technical indicators trying to describe the performance of the equity market. Stock technical indicators are indicators typically based on historical performance of the stock as well as the traded volume of that stock. There is an ever increasing amount of technical indicators in the existing literature which can generate contradictory signals. We used 35 commonly used stock technical indicators (see Table 2) extracted from the database Bloomberg.

Table 2. Technical indicators. C_t denotes the current price and L_t and H_t are the minimum and maximum prices in period t and MA means moving average. The exact formulas for some of the indexes are proprietary with the indicators for specific stocks obtainable from databases such as Bloomberg. Source: Blomberg, Kim [35].

N	Indicator	Description
1	SMAVG (50)	50 days simple MA. $\frac{\sum_{i=0}^{49} C_{t-i}}{50}$
2	SMAVG (100)	100 days simple MA. $\frac{\sum_{i=0}^{99} C_{t-i}}{100}$
3	SMAVG (200)	200 days simple MA. $\frac{\sum_{i=0}^{199} C_{t-i}}{200}$
4	Volume	Total traded volume
5	Stochastic %K (14-day)	$\frac{C_t - min(L_{t-n,\forall n \in \{1,\ldots,14\}})}{max(H_{t-n,\forall n \in \{1,\ldots,14\}}) - min(L_{t-n}, \forall n \in \{1,\ldots,14\})}$
6	%D (3-day)	3 days MA of stochastic %K. $\frac{\sum_{i=0}^{2} K_{t-i}}{3}$
7	%D (5-day)	5 days MA of stochastic %K. $\frac{\sum_{i=0}^{4} K_{t-i}}{5}$
8	Slow %D (3-day)	MA of the 3 days %D. $\frac{\sum_{i=0}^{2} D_{t-i}}{3}$
9	Slow %D (5-day)	MA of the 5 days %D. $\frac{\sum_{i=0}^{4} D_{t-i}}{5}$
10	Momentum (10-day)	Change in price over a 10 days period. $C_t - C_{t-10}$
11	Moving average (5D) of Momentum (10-day)	5 days MA of 10-day Momemtum. $\frac{\sum_{i=0}^{4} M_{t-i}}{5}$
12	ROC (daily)	Change in price in % over 1 day period. $\frac{C_t}{C_{t-1}}100$
13	Moving average (5D) of ROC	5 days MA of ROC. $\frac{\sum_{i=0}^{4} ROC_{t-i}}{5}$
14	Williams (14)	14 days Williams' indicator. $\frac{max(H_n, \forall n \in \{1,\ldots,14\}) - C_t}{max(H_n, \forall n \in \{1,\ldots,14\}) - min(L_n, \forall n \in \{1,\ldots,14\})}$
15	A/D oscillator	Accumulation distribution indicator. $\frac{H_t - C_{t-1}}{H_t - L_t}$
16	MA (5)	5 days moving average. $\frac{\sum_{i=0}^{4} C_{t-i}}{5}$
17	MA (10)	10 days moving average. $\frac{\sum_{i=0}^{9} C_{t-i}}{10}$
18	Disparity 5	$\frac{C_t}{MA_t}$
19	Disparity 10	$\frac{C_t}{MA_t}$
20	OSCP	$\frac{MA_t - MA_n}{MA_t}$
21	CMCI (13)	$\frac{(H_t+L_t+C_t)/3 - \frac{\sum(H_{t-i+1}+L_{t-i+1}+C_{t-i+1})}{3n}}{0.015(\frac{\sum\lvert H_{t-i+1}+L_{t-i+1}+C_{t-i+1}-\sum((H_t+L_t+C_t)/3)\rvert}{n}}$)
22	RSI (14) on Close	Relative strength index. $100 - \frac{100}{1+\frac{Average\ up}{Average\ down}}$
23	MA (250)	250 days simple MA. $\frac{\sum_{i=0}^{249} C_{t-i}}{250}$
24	Z-score	Distance in standard deviations from MA
25	AMAVG(14,2,30)	Exponential MA
26	EMAVG (5) on Close	5 days exponential moving average with exponential term $\frac{2}{t+1}$
27	TMAVG (5) on Close	5 days triangular moving average with factor $\frac{t+1}{2}$
28	VMAVG (5) on Close	Variable MA with smoothing based in volatility
29	MACD(12,26)	T_1 exponential MA (fast) $-$ T_2 exponential MA (slow)
30	Hurst(25)	Exponential MA with mean reversion assumption.
31	FG(5)	Ratio of buying and selling strength (5 days)
32	Kairi (Simple,25)	Deviation of prices from its MA
33	Elder Force Index	Relationship between price and trading volume movements
34	JKHL Index	Relationship between new high and lows smoothed by two MA.
35	RMI(Close,14,10)	Relationship between overbought and oversold levels

The main hypothesis is that it is possible to construct an algorithm that estimates an appropriate combination of inputs (technical indicators) for non-linear forecasting tools, generating lower error rates when forecasting the direction of stock movements than using directly all the technical indicators available. Another hypothesis is that on average such algorithm generates lower error rates than completely random combinations of the input variables (technical indicators).

The proposed approach was implemented in Matlab using neural networks as classifiers. 100 initial configurations times 2500 iterations were carried out for each index, which translates in 250,000 neural networks per index and a total of 2,000,000 neural networks (8 indexes). The neural networks used were back propagation classification neural networks with one hidden layer with 25 neurons and trained with the Levenberg-Marquardt rule. The value for the number of neurons was chosen as a result of the preliminary sensitivity analysis on the number of neurons presented in Section 3.2.

3.2. Results

Previously to testing the proposed strategy, a preliminary sensitivity analysis on the number of neurons has been carried out to find the most convenient number of neurons in the hidden layer. The full set of 35 technical indicators was used, and the number of neurons in the hidden layer was increased from 25 to 25,000 in steps of 25 neurons. Simply increasing the number of neurons did not appear to increase the classification accuracy of neural networks for the performance on stock index in the following period ($t+1$) (see Figure 2) for any of the six indexes analyzed.

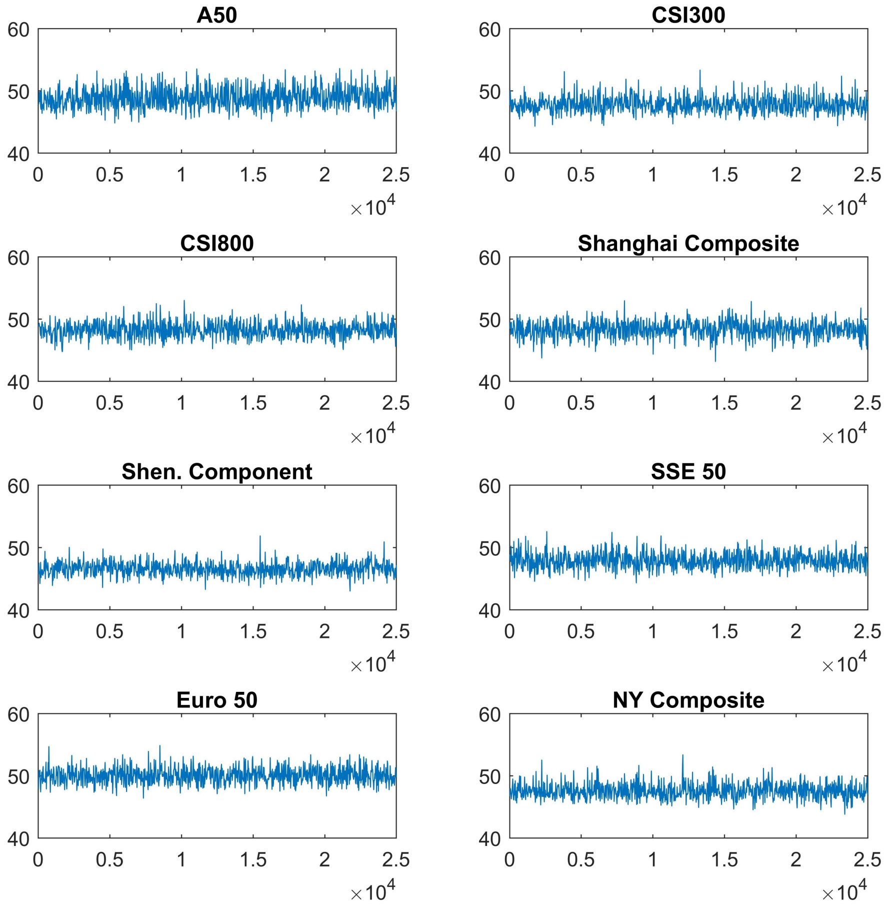

Figure 2. Sensitivity analysis of the error rate (using directly neural networks) as the number of neurons is increased.

On the other hand, Figure 3 shows an example of the evolution of the error rate (misclassification of up/down days) in the training of one the neural networks. It can be seen that training improves the fitting to the data up to a significant number of training iterations, showing that the NN is really learning the ground truth function ϕ.

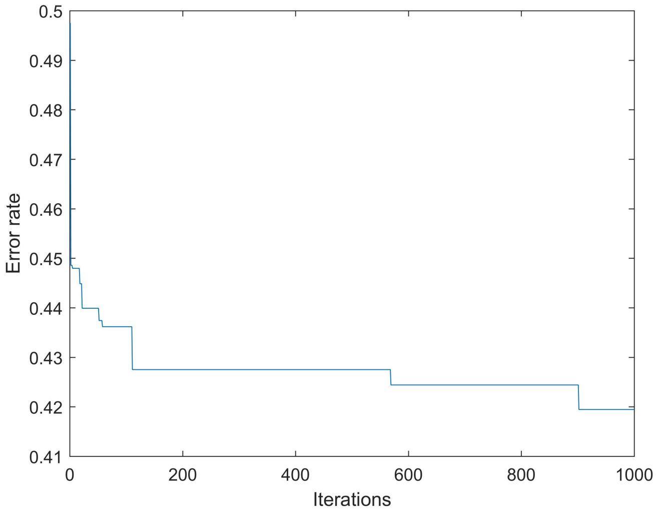

Figure 3. Example of the evolution of the error rate as the number of iterations increases.

The indicator selection approach is a combinatorial approach that can be used to select an appropriate combination of variables in non-linear models, using techniques such as neural networks. In the example illustrated in this article the proposed approach was implemented in six different Chinese stock indexes plus two world indexes. The error rate obtained using the algorithm in combination with neural networks was lower than the error rate obtained using directly neural networks including all the 35 available variables, see Table 3. The average improvement in the error rate (over all the considered indexes) was 9.1%. It turns out that this is a very good result considering how difficult is the task of predicting the movements of the market and the great amount of benefits that can be realized even with a small improvement in the forecasting. Moreover, in some of the indexes the improvement is quite high, in excess of 11% (A50, Shanghai Composite and SSE50). A very interesting finding is that the baseline approach, that is, considering all the technical indicators available for forecasting yields error rates greater than 50% in some cases (A50, CSI800, Shanghai Composite, SSE 50 and Euro Stoxx 50). This means that tossing a coin produces better results than using the full pool of indicators. The reason for that is that, as pointed out in Section 3.1, some of the technical indicators produce contradictory signals.

Table 3. Classification errors (up and down index movements) of the proposed approach using neural networks compared to the direct application of neural networks using all the 35 technical indicators.

Index	Direct (Error %)	Proposed (Error %)	Improv. (%)	Combinations
A50	54.0	41.2	12.8	(1,2,4,6,7,10,12,13,14,15,19,21,23,25,34,35)
CSI 300	46.9	40.3	6.6	(2,3,4,7,8,11,12,13,15,19,21,25,26,29,34)
CSI 800	50.7	41.1	9.6	(2,4,5,6,9,12,13,16,17,20,22,28,30,32,33,34)
Shanghai Composite	52.3	40.1	12.2	(1,2,4,5,6,9,10,12,13,15,16,18,19,20,21,22,24,33)
Shenzhen Component	45.6	40.7	4.9	(2,4,5,8,10,12,14,19,21,22,25,28,29,30,33,34,35)
SSE 50	51.0	39.6	11.4	(1,2,3,7,9,10,11,12,20,21,22,24,25,27,28,30,32,35)
Euro Stoxx 50	50.1	41.8	8.3	(3,4,7,11,17,18,20,22,25,27,28,31,33,34)
New York Composite	47.2	40.7	9.0	(1,3,7,8,9,11,14,15,17,19,20,25,27,28,29,33,34)
Average	49.7	41.0	9.1	

The histogram indicating the frequency of appearance of the technical indicators in the output of the algorithm for the various indexes shown in Table 3 can be seen in Figure 4. It is evident that some of the technical indicators are picked more frequently than others, thus they are more likely to provide better accuracy in the prediction of the price change direction. Nevertheless, this does not imply that using only these more relevant indicators will lead to a better forecasting. As an example the combination formed by the indicators with highest frequency of ocurrence in Table 3 and Figure 4, i.e., $\{2, 4, 12, 25, 34\}$, gives an average improvement of 2.3% over the baseline approach, which is significantly worse than that achieved by using the proposed combinations.

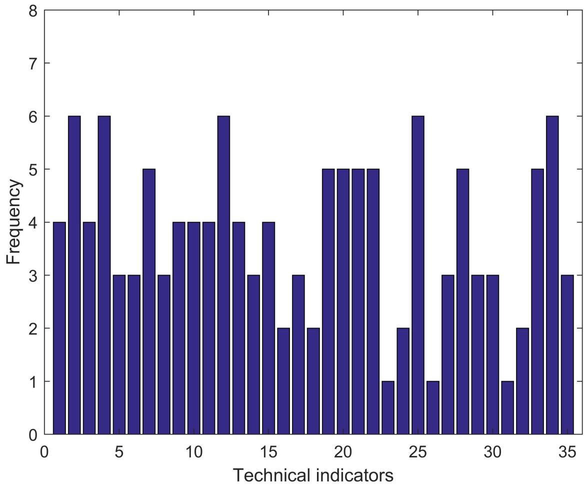

Figure 4. Histogram of the technical indicators appearance in the output of the proposed algorithm.

Table 4 shows the improvement along the iterations on the algorithm. The average improvement from the first iteration to the last was 7.1%. While this value clearly evinces that the algorithm improves the initial combinations, a more rigorous test has been done. A Wilcoxon test was carried out comparing the distribution of error rates obtained in the first and last iterations for each index. The Wilcoxon test rejects the hypothesis, for all the indexes analyzed, that the median of the error rates for the initial and final distributions are statistically the same (Table 5), suggesting that the iterative process does significantly improve accuracy. The same approach was followed to compare the error rate using neural networks directly (with all 35 technical indicators) with the error rate obtained using the technical indicator selection approach (Table 6). The Wilcoxon test rejects the null hypothesis that the median of the error rates obtained using these two methods are statistically equivalent, suggesting that the proposed method statistically significantly improved the forecasting accuracy of up/down stock index movements.

Table 4. Classification errors (up and down index movements) of the selection approach using neural networks.

Index	Error (Initial Iteration) %	Error (Final Iteration) %	Improvement %
A50	48.4	41.2	7.2
CSI 300	48.1	40.3	7.8
CSI 800	48.3	41.1	7.2
Shanghai Composite	46.9	40.1	6.8
Shenzhen Component	46.1	40.7	5.4
SSE 50	47.3	39.6	7.7
Euro Stoxx 50	49.7	41.8	7.9
New York Composite	47.5	40.6	6.9
Average	47.5	40.7	7.1

Table 5. Wilcoxon test comparing the error rate in the first and last iterations (*p*-values).

Index	*p*-Value
A50	5.5467×10^{-127}
CSI 300	6.3542×10^{-31}
CSI 800	2.8497×10^{-126}
Shanghai Composite	6.4421×10^{-115}
Shenzhen Component	7.8451×10^{-51}
SSE 50	6.5487×10^{-121}
Euro Stoxx 50	4.6587×10^{-114}
New York Composite	7.6257×10^{-112}

Table 6. Wilcoxon test comparing the error rate using neural networks directly with all the 35 technical indicators and the algorithm output.

Index	*p*-Value
A50	6.7894×10^{-172}
CSI 300	5.6284×10^{-161}
CSI 800	3.5487×10^{-155}
Shanghai Composite	4.6672×10^{-167}
Shenzhen Component	3.4837×10^{-175}
SSE 50	4.45728×10^{-156}
Euro Stoxx 50	4.6524×10^{-156}
New York Composite	2.7845×10^{-165}

Remark 2 has been also taken into account, and the algorithm has also been used with $M = 2$ and a total 125, 000 iterations which should be roughly equivalent to the parameters used previously, that is 100 combinations and 2500 iterations. The average improvement in this case was 8.7% which is marginally worse than that previously achieved. The difference could be due to the lower diversity of the set of candidate solutions, but also to the randomized nature of the strategy.

The average total time per index (100 initial configurations times 2500 iterations) was 157,691 s. The calculation time for each index can be seen in Table 7. The calculations were carried out in Matlab 2016 in an Intel, i5-3470, 3.2 GHz, 64 bit computer. The selection approach requires a significant amount of computation time but it clearly is faster than calculating all the possible combinations of technical indicators, which is for the example presented in this paper is not a feasible calculation in a normal computer.

Table 7. Calculation time.

Index	Total Time (sec)
A50	159,132
CSI 300	177,876
CSI 800	144,195
Shanghai Composite	150,710
Shenzhen Component	169,498
SSE 50	156,739
Euro Stoxx 50	142,816
New York Composite	160,566
Average	157,691

4. Discussion

The proposed method can be a feasible approach when trying to determine a combination of variables or features to be used when forecasting the behaviour of non-linear processes. In the particular example of the stock market there is a very large number of technical indicators that are intended to give the investor some indication of the future performance of the stock. These indicators can generate contradictory signals and selecting the appropriate combination of technical indicators can become a difficult tasks. Reducing the dimensionality of the problem is also important to avoid issues such as local minimum that can cause poor generalization when applying techniques such as neural networks. We showed in this paper that it is possible to use our approach in the Chinese stock market (generating an appropriate combination of independent variables for non-linear models) obtaining better results than directly applying neural networks to all the available independent variables. This was tested using 6 Chinese stock index (as well as two international indexes) and 35 technical indicators.

There was an average 9.1% improvement when using the combinatorial approach with neural networks over the results using directly all the technical indicators and neural networks as the non-linear forecasting technique. The formal statistical analysis comparing the results using neural networks directly (all technical indicators) with the results from the combinatorial approach using neural networks shows that there are statistically significant difference for the error rates obtained at a 1%, 5% and 10% significance level, supporting once more the hypothesis that the combinatorial approach using neural networks is a more appropriate tool for forecasting the direction of the stock market movement, at least for the 8 indexes analyzed, than using neural networks directly. Thus, for eight different indexes, better combinations of the technical indicators have been found, offering a practical choice to improve the forecasting accuracy and hence the expected benefits. The total calculation time per index (100 initial configurations time 2500 iterations) was 157,691. While this is a substantial amount of time it is a calculation that can be done with a normal laptop computer. Moreover, many of the operations of the proposed algorithm can be done in parallel further shortening the computation times.

While direct comparison is challenging the approach of using the combinatorial approach with neural networks seems to be generate better results for stock forecasting purposes than other approaches used in the existing literature such as for instance the Box–Jenkins approach used by Groda and Vrbka [3], which the authors considered not suitable. A more comparable paper is Kim and Han [36] that achieved a hit rate of 61% in the Korean market using genetic algorithm in combination to neural networks which is comparable to the 59% rate that we obtained in the Chinese market. Nevertheless comparison across different stock markets should be taken with caution. For instance, it should be naïve to believe that the same approach would generate the same results in two markets as different as South Korea and China with China being an open stock market dominated by institutional investors while the Chinese market is a market dominated by local retail investors.

The selection approach was illustrated in the context of the stock market and using neural networks but the approach is easily applicable to other fields. This is increasingly important as the amount of data available in many fields has increased substantially over the last few decades with an ever increasing need for tools to process large databases. Besides neural networks other non-linear models, such as for instance support vector machines, can be used using the proposed approach. This could be an interesting area of future work.

5. Conclusions

The proposed combinatorial method for variable selection is applicable to the problem of forecasting the direction of stock market movements using non-linear techniques such as neural networks. This approach generates better results than directly applying non-linear forecasting methods such as neural networks using all the available variables. For large amount of technical indicators (independent variables) it is clearly not possible to estimate the forecasts for all the combinations with the proposed method providing a reasonable alternative. In fact, it has been shown that using all the available indicators can be counterproductive as it has higher error than a pure random approach. Another relevant result is that better indicator choices have been researched for 8 different indexes. The calculation time for the combinatorial approach is another factor to take into account as it is a computationally demanding, but clearly more efficient, from a calculation time point of view, than estimating the forecasts for all the possible combinations.

The combinatorial approach was tested thoroughly for the Chinese stock market as well as for some indexes describing the US and European stock market. The forecasting accuracy of this approach in other markets might be different and this can be potentially an area of future work. There are several factors that could potentially impact the forecasting accuracy of this approach. For instance, narrow and deep markets might have different behaviors and hence the combinatorial approach might also have rather different forecasting accuracy.

Overall, when there are many potential variables (technical indicators) that can affect the performance of the stock market and no strong fundamental support for choosing a specific combination of these variables, the proposed approach can be an appropriate alternative. Similarly, while the approach was tested using neural networks it can be easily applied to other non-linear forecasting techniques such as support vector machines. It can also be generalized to other forecasting problems besides stocks. It is in that sense a rather general approach with many potential applications.

Author Contributions: Conceptualization, G.A. and D.R.R.; methodology, G.A.; software, G.A. and D.R.R.; validation, G.A. and D.R.R.; formal analysis, G.A. and D.R.R.; investigation, G.A. and D.R.R.; resources, G.A. and D.R.R.; data curation, G.A.; writing—original draft preparation, G.A.; visualization, G.A. and D.R.R.; supervision, G.A. and D.R.R.; project administration, G.A. and D.R.R.; funding acquisition, D.R.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received support by Ministerio de Ciencia e Innovación of Spain under project PID2019-106212RB-C41.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vrbka, J.; Rowland, Z. Stock price development forecasting using neural networks. In *SHS Web of Conferences—Innovative Economic Symposium 2017: Strategic Partnership in International Trade*; EDP Sciences: Les Ulis, France, 2017. [CrossRef]
2. Horak, J.; Krulicky, T. Comparison of exponential time series alignment and time series alignment using artificial neural networks by example of prediction of future development of stock prices of a specific company. In *SHS Web of Conferences: Innovative Economic Symposium 2018—Milestones and Trends of World Economy (IES2018)*; EDP Sciences: Les Ulis, France, 2018. [CrossRef]
3. Groda, B.; Vrbka, J. Prediction of stock price developments using the Box–Jenkins method. In *SHS Web of Conferences—Innovative Economic Symposium 2017: Strategic Partnership in International Trade*; EDP Sciences: Les Ulis, France, 2017. [CrossRef]
4. Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
5. Hocking, R.R. A Biometrics Invited Paper. The analysis and selection of variables in linear regression. *Biometrics* **1976**, *3*, 1–49. [CrossRef]
6. Yuan, M.; Lin, Y. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B* **2006**, *68*, 49–67. [CrossRef]
7. Rech, G.; Terasvirta, T. A simple variable selection technique for nonlinear models. *Commun. Stat. Theory Model.* **2000**, *30*, 1227–1241. [CrossRef]
8. Ye, M.; Sun, Y. Variable Selection Via Penalized Neural Networks: A Drop-Out-One Loss Approach. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018; Volume 28, pp. 5620–5629.
9. Qin, Q.; Wang, Q.-G.; Ge, S.G.; Ramakrishnan, G. Chinese Stock Price and Volatility Predictions with Multiple Technical Indicators. *J. Intell. Learn. Syst. Appl.* **2011**, *3*, 209–219. [CrossRef]
10. Paskalis, G. Market Timing with Moving Averages. In Proceedings of the 25th Australasian Finance and Banking Conference, Sydney, Australia, 16–18 December 2012.
11. Bruni, R. Stock Market Index Data and indicators for Day Trading as a Binary Classification problem. *Data Brief.* **2016**, *10*, 569–575. [CrossRef]

12. Stankovic, J.; Stojanovic, M. Investment strategy optimization using technical analysis and predictive modeling in emerging markets. *Procedia Econ. Financ.* **2015**, *19*, 51–62. [CrossRef]
13. Souza, M.J.S.; Ramos, D.G.F.; Pena, M.G. Examination of the profitability of technical analysis based on moving average strategies in BRICS. *Financ. Innov.* **2018**, *4*. [CrossRef]
14. Qi, L. Technical analysis and stock return predictability: An aligned approach. *J. Financ. Mark.* **2018**, *38*, 103–123. [CrossRef]
15. Jian, W.; Junseok, K. Predicting Stock Price Trend Using MACD Optimized by Historical Volatility. *Math. Probl. Eng.* **2018**, *18*. [CrossRef]
16. Brown, D.; Jennings, R. On Technical Analysis. *Rev. Financ. Stud.* **1989**, *2*, 527–551. [CrossRef]
17. Ramazan, G. The predictability of security returns with simple technical trading rules. *J. Empir. Financ.* **1998**, *5*, 347–359.
18. Bokhari, J.; Cai, C.; Hudson, R. The predictive ability and profitability of technical trading rules: Does company size matter? *Econ. Lett.* **2005**, *86*, 21–27. [CrossRef]
19. Park, C.; Scott, I. What do we know about the profitability of technical analysis? *J. Econ. Surv.* **2007**, *21*, 786–826. [CrossRef]
20. Kimoto, T.; Asakawa, K.; Yoda, M.; Takeoka, M. Stock Market Prediction System with Modular Neural Networks. *IJCNN Int. Jt. Conf. Neural Netw.* **1990**, *1*, 1–6. [CrossRef]
21. Yoon, Y.; Swales, G. Predicting stock price performance: A neural network approach. In Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences, Kauai, HI, USA, 8–11 January 1991; Volume 4, pp. 156–162. [CrossRef]
22. Zhang, G. A neural network ensemble method with jittered training data for time series forecasting. *Inf. Sci.* **2007**, *177*, 5329–5346. [CrossRef]
23. Mahdi, P.; Hamidreza, T. Stock market value prediction using neural networks. In Proceedings of the 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM), Krackow, Poland, 8–10 October 2010; Volume 177, pp. 132–136.
24. Chen, J.; Chen, W.; Huang, C. Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks. In Proceedings of the 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, China, 16–18 November 2016; Volume 4, pp. 87–92. [CrossRef]
25. Qiu, J.; Wang, B.; Zhou, C. Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE* **2016**, *15*. [CrossRef]
26. Baldi, P.; Hornik, K. Neural Networks and Principal Component Analysis: Learning from Examples without Local Minima. *Neural Netw.* **1989**, *2*, 53–58. [CrossRef]
27. Gori, M.; Tesi, A. On the problem of local minima in backpropagation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 76–86. [CrossRef]
28. Bianchini, M. On the problem of local minima in recurrent neural networks. *IEEE Trans. Neural Netw.* **1994**, *5*, 167–177. [CrossRef]
29. Yu, X. On the local minima free condition of backpropagation learning. *IEEE Trans. Neural Netw.* **1995**, *6*, 1300–1303. [CrossRef] [PubMed]
30. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
31. Wang, J.; He, H. A folded neural network autoencoder for dimensionality reduction. *Procedia Comput. Sci.* **2012**, *13*, 120–127. [CrossRef]
32. Becker, M.; Zielke, T. Robust dimensionality reduction for data visualization with deep neural networks. *Graph. Model.* **2020**, *108*. [CrossRef]
33. Kiarashinejad, Y.; Abdollahramezani, S. Deep Learning Approach Based on Dimensionality Reduction for Designing Electromagnetic Nanostructure. *Comput. Mater.* **2020**, *6*. [CrossRef]
34. Breger, A. Supervised learning and dimension reduction techniques for quantification of retinal fluid in optical coherence tomography images. *Eye* **2017**, *31*, 1212–1220. [CrossRef]

35. Kim, K. Financial Time Series Forecasting Using Support Vector Machines. *Neurocomputing* **2003**, *55*, 307–319. [CrossRef]
36. Kim, K.; Han, L. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Syst. Appl.* **2000**, *19*, 125–132. [CrossRef]