

CAST: Using neural networks to improve trading systems based on technical analysis by means of the RSI financial indicator

Alejandro Rodríguez-González

alejandro.rodriguez@uc3m.es
ComputerScienceDepartment
Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, SPAIN
Phone: +34 91 624 5936
Fax: +34 91 624 9129

Ángel García-Crespo

angel.garcia@uc3m.es
ComputerScienceDepartment
Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, SPAIN
Phone: +34 91 624 9417
Fax: +34 91 624 9129

Ricardo Colomo-Palacios (CorrespondingAuthor)

ricardo.colomo@uc3m.es
Computer Science Department
Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, SPAIN
Phone: +34 91 624 5958
Fax: +34 91 624 9129

Fernando Guldrís Iglesias

fernando.guldris@uc3m.es
ComputerScienceDepartment
Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, SPAIN
Phone: +34 91 624 5936
Fax: +34 91 624 9129

Juan Miguel Gómez-Berbís

juanmiguel.gomez@uc3m.es
Computer Science Department
Universidad Carlos III de Madrid
Av. Universidad 30, Leganés, 28911, Madrid, SPAIN
Phone: +34 91 624 5958
Fax: +34 91 624 9129

ABSTRACT

Stock price predictions have been a field of study from several points of view including, among others, artificial intelligence and expert systems. For short- term predictions, the technical indicator Relative Strength Indicator (RSI) has been published in many papers and used worldwide.

CAST is presented in this paper. CAST can be seen as a set of solutions for calculating the RSI using artificial intelligence techniques. The improvement is based on the use of feedforward neural networks to calculate the RSI in a more accurate way, which we call the iRSI. This new tool will be used in two scenarios. In the first, it will predict a market-- in our case, the Spanish IBEX 35 stock market. In the second, it will predict single-company values pertaining to the IBEX 35. The results are very encouraging and reveal that the CAST can predict the given market as a whole along with individual stock pertaining to the IBEX 35 index.

Keywords: Neural networks; Generalized Feedforward; Technical Analysis; Relative Strength Index; Ibex 35

1. Introduction

There has been growing interest in decision support trading systems in recent years. Forecasting price movements in stock markets has been a major challenge for common investors, businesses, brokers and speculators (Majhi, Panda & Sahoo, 2009). The stock market is considered a highly complex and dynamic system with noisy, non-stationary and chaotic data series (Wen et al., 2010), and hence, difficult to forecast (Oh & Kim, 2002; Wang, 2003). However, in spite of its volatility, it is not entirely random (Chiu & Chen, 2009). Instead, it is non-linear and dynamic (Abhyankar, Copeland, & Wong, 1997; Hiemstra & Jones, 1994) or highly complicated and volatile (Black & Mcmillan, 2004). Stock movement is affected by the mixture of two types of factors (Bao & Yang, 2008): determinant (e.g., gradual strength change between buying side and selling sides) and random (e.g., emergent affairs or daily operation variations).

According to Wen et al. (2010), the study of the stock market is a hot topic, because if successful, the result will transfer to fruitful rewards. Thus, it is obvious that predicting the stock market's movement is the long-cherished desire of investors, speculators, and industries (Kim, 2004). However, this market is extremely hard to model with any reasonable accuracy (Wang, 2003). Prediction of stock price variation is a very difficult task and price movement behaves more like a random walk and time varying (Chang & Liu, 2008).

However, in spite of this complexity, many factors, including macroeconomic variables and stock market technical indicators, have been proven to have a certain level of forecast capability in the stock market during a certain period of time (Lo, Mamaysky, & Wang, 2000). One of the tools for this financial practice is technical analysis, also known as "charting". According to Leigh et al. (2002), Charles Dow developed the original Dow Theory for technical analysis in 1884 revisited by Edwards and Magee (1997) more than a century earlier. Technical analysis studies historical data surrounding price and volume movements of the stock by using

charts as the primary tool to forecast future price movements (Murphy, 1999). In recent years, and in spite of several critics (E.g. Malkiel, 1995), technical analysis has proven to be powerful for evaluating stock prices and is widely accepted among financial economists and brokerage firms (Chavarnakul&Enke, 2008).

Due to this importance, a lot of research has gone into the development of models based on a range of intelligent soft computing techniques over the last two decades (Majhi, Panda &Sahoo, 2009). Most of the work is the combination of soft computing technology and technical analysis in stock analysis (Wen et al., 2010; Chen et al., 2009).

Following this research trend, in this paper, CAST is presented. CAST is a tool designed to improve the investment techniques used in trading systems, applied to the Spanish stock market, based on a new way to calculate the Relative Strength Index (RSI) by Wilder (1978). This improvement is based on the use of feedforward neural networks to calculate RSI in a more accurate way, which we call iRSI.

The paper consists of five sections and is structured as follows. Section 2 reviews the relevant literature about technical analysis and its intersection with soft computing. Section 3 discusses the main features of CAST, including the conceptual model, algorithm and architecture. Section 4 describes the evaluation of the tool's performance including a description of the sample, the method, results and discussion. Finally, the paper ends with a discussion of research findings, limitations and concluding remarks.

2. Background

Stock price prediction using soft intelligence methods is not new. To solve the non-linear problem and improve stock price evaluation, many researchers have focused on technical analysis and used advanced maths and science (Wang & Chan, 2006). Along with the development of artificial intelligence, more and more researchers try to build automatic decision-making systems to predict the stock market (Kovalerchuk&Vityaev, 2000). Soft computing techniques such as fuzzy logic, neural networks, and probabilistic reasoning draw most attention because of their abilities to handle uncertainty and noise in the stock market (Vanstone & Tan, 2005). However, though soft computing can somewhat reduce the impact of random factors, low-level data are so uncertain that they even behave purely randomly at some time (Peters, 1994). More in depth, neural networks have also become an important method for stock market prediction because of their ability to deal with uncertain, fuzzy, or insufficient data that fluctuate rapidly in very short periods of time (Schoeneburg, 1990). Furthermore, neural networks are able to decode non-linear time series data that adequately describe the characteristics of the stock markets (Yao, Tan, &Poh, 1999), and can be applied to various complex financial markets directly (Roh, 2007). Thus, banks and financial institutions are investing heavily in the development of neural network models and have started to deploy them in the financial trading arena. Their ability to 'learn' from the past and produce a generalized model to forecast future prices, freedom to incorporate fundamental and technical analysis into a forecasting model and ability to adapt according to market conditions are some of the main reasons for their popularity (Majhi, Panda &Sahoo, 2009).

White (1988) was the first to use neural networks for market forecasting in the late 1980s. In the early 1990s, Kimoto, Asakawa, Yoda, and Takeoka (1990) used several learning algorithms and prediction methods to develop a prediction system for the Tokyo Stock Exchange Prices Index. Trippi and DeSieno (1992) combined the outputs of individual networks using Boolean operators to produce a set of composite rules. Other artificial neural network approaches can be found in various papers from that decade (e.g., Aiken & Bsat, 1994; Austin & Looney, 1997; Brownstone, 1996; Chenoweth, Obradovic & Stephenlee, 1996; Grudnitski & Osburn, 1993; Saad, Prokhorov & Wunsch, 1998; Thammano, 1999; Yoon & Swales, 1991). According to Chang & Liu (2008), however, these models have their limitations owing to the tremendous noise and complex dimensionality of stock price data, and besides, the quantity of data itself and the input variables may also interfere with each other.

Recently, in the first decade of the 21st century, various studies using ANN have been developed in the fields of forecasting stock indexes (Chang et al., 2009; Chavarnakul & Enke, 2008; Chen, Leung, & Daouk, 2003; Chen & Leung, 2004; Enke & Thawornwong, 2005; Lam, 2004; Lee & Chen, 2002; Lee & Chiu, 2002; Leigh, Hightower & Modani, 2005; Thawornwong & Enke, 2004; Yao, Li & Tan, 2000). The importance of further developments in soft computing led to several papers devoted to forecasting stock indexes using techniques such as support vector machines (e.g., Chiu & Chen, 2009; Huang, Nakamori, & Wang, 2005; Kim, 2003; Pai & Lin, 2005; Wen et al., 2010), fuzzy systems (e.g., Chang & Liu, 2008; Chang, Wang & Liu, 2007; Huang & Yu, 2005; Wang, 2003), genetic algorithms (e.g., Chen et al., 2007; Chen et al., 2009; Oh, Kim & Min, 2005; Oh et al., 2006; Potvin, Soriano & Vallee, 2004) and mixed methods (e.g., Armano, Marchesi & Murru, 2005; Armano, Murru & Roli, 2002; Chen, Chen & Lin, 2010; Hassan, Nath & Kirley, 2007; Leigh, Purvis & Ragusa, 2002; Kwon & Moon, 2007).

As stated before, the CAST (Chartist Analysis System For Trading) is based on the use of an improved version of the RSI, one of the leading technical analysis indexes. The RSI as a part of diverse calculations and formulas is commonly present in soft computing research (e.g., Chang & Liu, 2008; Chang et al., 2009; Chiam, Tan & Al Mamum, 2009; Chiu & Chen, 2009; Kim, 2004; Lai et al., 2009; Lu, Lee & Chiu; Majhi, Panda & Yahoo, 2009; Tan, Queck & Yow, 2008; Yao & Herbert, 2009). However, using soft computing methods in getting iRSI calculations is a research task with no presence in the literature. In this paper is proposed CAST, a system that uses a generalized feedforward neural network to perform improved RSI calculations.

3. System Core

The idea of the system developed in the CAST is to create a trading system based on fundamental or chartist analysis. Concretely, the main idea is to use one of the most used financial indicators, namely, the RSI. As described before, the RSI is a financial technical analysis momentum oscillator measuring the velocity and magnitude of directional price movement by comparing upward and downward close-to-close movements. Momentum measures the rate of the rise or fall in stock prices. Is the momentum increasing in the "up" direction, or is the momentum increasing in the "down" direction.

There are several ways to calculate this indicator, and it depends on whether you want to calculate a "normal RSI" or gentler RSI formulas. The calculation of the RSI is described as follows:

For each day, an upward change (U) or downward change (D) is calculated. "Up" days are characterized by the daily close being higher than yesterday's daily close, i.e.,:

$$U = close_{today} - close_{yesterday}$$

$$D = 0$$

Conversely, a down day is characterized by the close being lower than the previous day's (note that D is nonetheless a positive number).

$$U = 0$$

$$D = close_{yesterday} - close_{today}$$

If today's close is the same as yesterday's, both U and D are zero. An average for U is calculated with an **exponential moving average** (EMA) using a given N-days smoothing factor, and likewise for D. The ratio of those averages is the Relative Strength:

$$RS = \frac{EMA[N]_{of} U}{EMA[N]_{of} D}$$

This is converted to a Relative Strength Index between 0 and 100:

$$RSI = 100 - 100 \times \frac{1}{1 + RS}$$

This value is the key field used in the system.

Wilder (1978) established that the most accurate value for value N to calculate the best RSI is 14 because it was half of the lunar cycle. However, depending on the market, the company and other factors, the value 14 is not always the best value to calculate the RSI.

For this reason, our first efforts on this topic tried to calculate a better value for N in order to calculate better RSI values that allows for making more reliable investments. Nevertheless, our research shows that calculating the value N with the neural network approach didn't give as good results as calculating an RSI value directly (explained in the evaluation section). With the purpose of improving this value, the CAST system was created.

In this paper, the CAST is described. It is a system capable of predicting RSI values for a concrete market instead of for a concrete company. The main idea is to try to predict market behavior, concretely, RSI behavior (as a collection of companies) and particularize for a concrete company using some correction factors. Figure 1 shows the general architecture of the system:

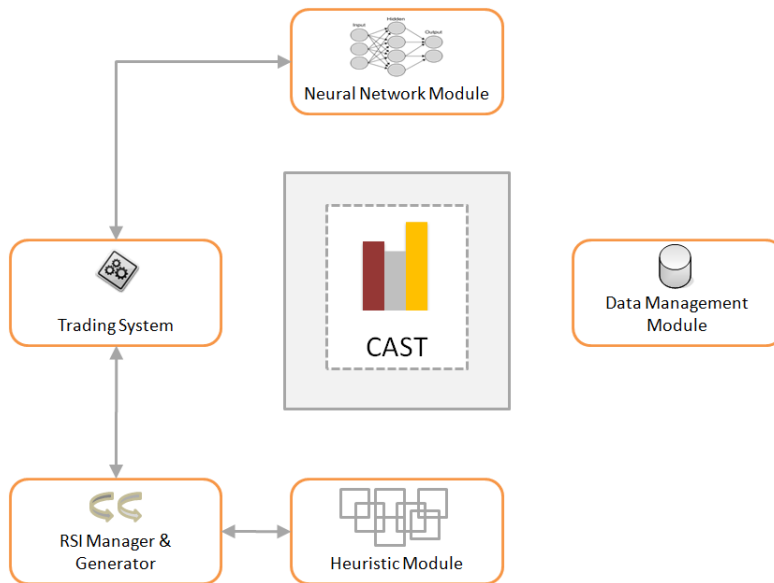


Figure 1. System architecture

In the following sections, all the modules presented in the architecture will be explained.

3.1. Neural Network Module

The Neural Network Module is responsible for providing the RSI values that will be used to decide if an investor should invest in a certain company. Figure 2 shows the representation (using NeuroSolutions software) of the neural network that was used in the final version of the system:

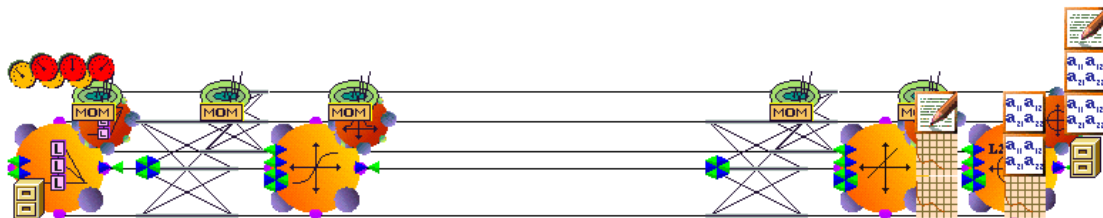


Figure 2. Representation of the network

The network used is a generalized feedforward network (Arulampalam&Bouzerdoum, 2003). Generalized feedforward networks are a special case of multilayer perceptrons such that connections can jump over one or more layers. In theory, an MLP can solve any problem that a generalized feedforward network can solve. In practice, however, the generalized feedforward networks often solve the problem much more efficiently (Fine, 1999). The advantage of the generalized FF network is in its ability to project activities forward by bypassing layers. The result is that the training of the layers closer to the input becomes much more efficient. Figure 3 shows the structure of this kind of network:

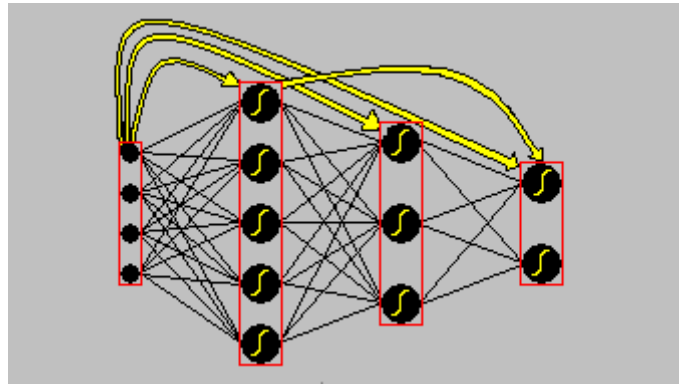


Figure 3. Representation of a generalized feedforward network

Now, the configuration of the network used will be shown:

In first place, the input values of the network are shown in Table 1:

Input Values	Explanation
IBEX35 Action Value	Value of the market for a concrete day.
RSI(9)	Calculus of RSI value using N = 9.
RSI(14)	Calculus of RSI value using N = 14.
RSI(30)	Calculus of RSI value using N = 30.
RSI Optimal	Calculus of optimal RSI using Heuristic (*)

Table 1. Network Input Values

(*): Heuristic explanation and calculation are shown in Section 3.5.

From the beginning, Wilder (1978) proposes the use of 14 as the optimal number of days to use the RSI. However, subsequent papers studied the possibility of using a different time interval. Pring (1991) pointed out that changing the period of days should be considered depending on whether you are looking for a short- term or long-term prediction. Therefore, he proposes using 9 days for very short-term predictions and 22 days for a longer period. Nevertheless, in practice, longer periods are used frequently, so authors decided to use 30 days as a representative longer period.

The topology of the network used was divided into an input layer, one hidden layer and an output layer. Table 2 shows the neurons set to each layer.

Layer	Input Neurons	Output Neurons	Activation Function
Input	5	15	Laguarre (3 Taps, 1 Tap Delay)
Hidden	15	10	TanH
Output	10	1	Bias

Table 2. Network Topology Values

For nearly all problems, one hidden layer is sufficient. Two hidden layers are required for modeling data with discontinuities. Using two hidden layers rarely improves the model, and it may introduce a greater risk of converging to a local minima. There is no theoretical reason for using more than two hidden layers (Cybenko, 1989). This is the reason to set number of hidden layers to one.

One of the most important characteristics of a perceptron network is the number of neurons in the hidden layer(s). If an inadequate number of neurons are used, the network will be unable to model complex data, and the resulting fit will be poor.

If too many neurons are used, the training time may become excessively long, and, worse, the network may overfit the data. When overfitting occurs, the network will begin to model random noise in the data. The result is that the model fits the training data extremely well, but it generalizes poorly to new, unseen data.

There are many rule-of-thumb methods for determining the correct number of neurons to use in the hidden layers (Murata et al, 1994), such as the following:

- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
- The number of hidden neurons should be less than twice the size of the input layer.

Nevertheless, the number of neurons that establish this kind of rules doesn't provide good results. For this reason, after testing some configurations, the number of neurons established was that shown in the table.

A typical neural network might have a couple of hundred weights whose values must be found to produce an optimal solution. If neural networks were linear models like linear regression, it would be easy to find the optimal set of weights. However, the output of a neural network as a function of the inputs is often highly non-linear; this makes the optimization process complex. For this reason, the aforementioned activation function has been used.

The Laguarre activation function shown in Table 2 is, in fact, an input layer used when processing temporal sequences. This allows the temporal signal to be presented directly to the network without preprocessing or segmentation.

The Laguarre memory structure is built from a low-pass filter with a pole at $z = (1-m)$, followed by a cascade of K all-pass functions. This provides a recursive memory of the input signal's past. Notice that the axon receives a vector of inputs. Therefore, the Laguarre implements a vector memory structure. The memory depth is equal to K/m , where K is the number of taps and is the Laguarre coefficient. The Laguarre coefficient is implemented by the axon's weight vector, i.e. $\mu = w_i$. This allows each PE to have its own coefficient, each of which can be adapted. The delay between taps, t , is an adjustable parameter of the component.

Tap Activation Function:

$$T^0(z, w_i) = \frac{\sqrt{1 - (1 - w_i)^2}}{1 - (1 - w_i)^{z^{-t}}} \quad T^k(z, w_i) = \frac{1 - (1 - w_i)}{z^{-t} - (1 - w_i)} \quad k > 0$$

The TanH function, also known as the Hyperbolic Tangent function, is expressed as:

$$\tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

The TanH activation method will be fed the sum of the input patterns and connection weights. This sum will be referred to as “u”. The TanH activation method simply returns the hyperbolic tangent of “u”. The hyperbolic tangent threshold method will be a range of numbers both greater than and less than zero.

The bias activation function is as follows:

$$f(x_i, w_i) = x_i + w_i$$

The backpropagation training algorithm was used to train the network. It was first described by Rumelhart and McClelland (1986); it was the first practical method for training neural networks. The original procedure used the gradient descent algorithm to adjust the weights toward convergence using the gradient. Because of this history, the term “backpropagation” or “backprop” is often used to denote a neural network training algorithm using gradient descent as the core algorithm. That is somewhat unfortunate since backward propagation of error information through the network is used by nearly all training algorithms, some of which are much better than gradient descent.

Backpropagation using gradient descent often converges very slowly or not at all. On large-scale problems, its success depends on user-specified learning rate and momentum parameters. There is no automatic way to select these parameters, and if incorrect values are specified, the convergence may be exceedingly slow, or it may not converge at all. While backpropagation with gradient descent is still used in many neural network programs, it is no longer considered to be the best or fastest algorithm.

As for the momentum, it needs to be said that it is probably the most popular extension of the backprop algorithm; it is hard to find cases where this is not used (Yu & Chen, 1996). With momentum m, the weight update at a given time t becomes:

$$\Delta\omega_{ij}(t) = \mu_i * \delta_i * y_j + m * \Delta\omega_{ij}(t-1) \quad 0 < m < 1.$$

where m is a new global parameter which must be determined by trial and error. Momentum simply adds a fraction m of the previous weight update to the current one. When the gradient keeps pointing in the same direction, this will increase the size of the steps taken towards the minimum. It is therefore often necessary to reduce the global learning rate μ when using a lot of momentum (m close to 1).

The momentum value associated with the layers of the networks was finally set to 0.7. Hereafter the training values used in the network are shown in Table 3:

Number of input data	Cross Validation	Training
4061 (> 16 Years of Financial Data)	20% of Input Data	10% of Input Data

Table 3. Network Training Values

The number of epochs to train the network was set to 10,000, establishing that the network must stop learning after 500 epochs without improvement in cross-validation (Picard & Cook, 1994) error. Figure 3 shows the learning curve that can be seen when the network learning starts to stabilize.



Figure 3. Learning Curve

3.2. Trading System

The Trading System of CAST architecture is the part in charge of analyzing the result given by the neural network module. When a consultation is made to the CAST system, it takes the current values of the market and makes a query to the neural network. It is important to emphasize that in this case the predictions made by the system affect only the IBEX35 market and not to a concrete company. However, in the heuristic section, some correction values will be shown in order to see that it is possible to adapt the heuristic method to make predictions directly about the companies. Nevertheless, this modification has only been tested directly taking the RSI of the heuristic instead of training a neural network.

The analysis made by the trading system is simple. It takes the value given by the neural network (RSI) and compares it with two extreme values.

- If the RSI value is higher than 70 the decision that the trading system will return is a sell signal. This value can be adapted and in some cases the value will be set to 65 instead of 70.
- If the RSI value is lower than 30 the decision that the trading system will return is a buy signal. This value can be adapted and in some cases the value will be set to 35 instead of 30.

Initially, Wilder proposes setting the values of lower and greater extremes to 30 and 70. However, it is normal that analysts observe the tendency of the indicator and delimit their predictions to values 40 and 60. As such, we will need to analyze the tendency graph, so we will need to consider the values 35 and 65 to move forward tendencies and buy and sell signals.

3.3. Data Management Module

The data management module in this case only contains a sub-module called Reader. This is the part of the system in charge of obtaining the data necessary to work with the system. In this part we are going to indicate two sets of values necessary for the system. The first set refers to the case when we are predicting only IBEX35 market values and we are not

particularizing in a concrete company. The second set refers to the case when we want to predict values for a concrete company, so, different values are needed.

The values necessary for the system when we are trading on the IBEX35 market are the following:

- Value of the Market
- Date of the value

The values necessary for the system when we are trading on a concrete company of the IBEX35 Market are the following:

- Value of the Market
- Value of the company
- Date of the values

The only difference between the two kinds of trading systems is a single value that represents the value of the action. These are the main values that the system needs. Other values needed will be calculated by the system from these values. The data used in the system can be obtained from many different open sources like Yahoo Finance, the Madrid Stock Exchange (IBEX35 Market), The Financial Times, Reuters or Google Finance, among others. That is why there is only a sub-module called Reader because we only need to read data and process it.

3.4. RSI Manager and generator

The RSI Manager and generator is the module in charge of managing and generating the RSI Values. The generation of these values follows the mathematical expressions mentioned at the beginning of Section 3.

This module will calculate RSI(N) values where N will be between 5 and 35 (both included) and that will be used to train the network or to query it. However, there is another RSI calculation that should be made: Optimal RSI. There are two ways to calculate Optimal RSI.

- Optimal RSI for Market (IBEX35) prediction: The calculation is provided by the heuristic function (see the next section).
- Optimal RSI for Company (IBEX35) prediction: In the context of company prediction Optimal RSI generation has two ways to calculate the value:
 - **Optimal RSI with No Memory:** The calculation of the RSI with the heuristic is made taking into account the values that actions have in the whole period without any modification, so partially calculated RSIs are not stored and the RSI from the day before and the current day will always be used without taking into account previous RSIs. See the next section to see the formula.
 - **Optimal RSI with Memory:** In this case, the procedure is the same but you start at a concrete date in the past (5 years, for example) and RSIs are calculated taking the previous calculated RSIs as a reference and saving a memory of them. See the next section to see the formula.

3.5. Heuristic module

The heuristic module is in charge of managing the different formulas that provide the heuristic used to generate the optimal values for the RSI indicator. As was mentioned in the previous section, there is more than one way to calculate the Optimal RSI value.

Optimal RSI for Market (IBEX35) prediction:

The formula of this heuristic is the following:

$$RSI = C1 + C2 * IBEX35 + C3$$

where:

- IBEX35 represents the current value of the market on the current date.
- C1 is a correction parameter set to -206.1103731082200000
- C2 is a correction parameter set to 0.0213265642264200
- C3 is a correction parameter set to 0.9947712774802800

The formula has been obtained using statistical techniques applied to the variables involved in the process. Concretely, a linear regression to relate the RSI values with IBEX35 stock market closing values has been done.

To be able to collect the possible values of the RSI obtained in function of the number of days used for their calculi, a total of 30 linear regressions had been done. Each of these regressions uses a different value of the RSI when calculated itself with different time intervals. Concretely, it was allowed that the number of days used in the calculi fluctuate between 5 and 35. The reason is to collect all the possible values used, because it is not normal that analysts use periods outside of this range.

Each of the linear regressions calculated includes a parameter called AR(1) to improve the specification of the model. Once results have been obtained, the models that didn't fulfill the following requisites were discarded:

- Some of the independent variables are not statistically significant for explaining the behavior of the dependent variable.
- The probability of whole nullity of the model is zero.
- The goodness of fit is not good enough. In this sense, the models that cannot explain a 75% of the variation of the dependent variable are considered not valid.
- The model presents heteroscedasticity.
- The model presents autocorrelation.

Finally, in function of the criterions mentioned, the models that include the RSI calculated with intervals of days equals to 5, 10, 12, 28, 30 and 35 have been discarded. With the rest of the models, we proceed to calculate a unique equation that is its arithmetic mean, so that the resulting heuristic is the most representative value of all the valid calculated models.

Optimal RSI for Company (IBEX35) prediction:

In this case, there are two ways to calculate the RSI value.

Optimal RSI with No Memory:

The formula of this heuristic is the following (explained in several steps):

$$rsiToday = C1 + (C2 * vMT) + C3$$

$$rsiYday = C1 + (C2 * vMY) + C3$$

$$restRSI = rsiToday - rsiYday$$

$$growthRateCompany = (vCT - vCY) / vCY$$

$$growthRateMarket = (vMT - vMY) / vMY$$

$$RSI = rsiYday + (restRSI * \frac{growthRateCompany}{growthRateMarket})$$

Where:

- vMT represents the value of the market today.
- vMY represents the value of the market yesterday.
- vCT represents the value of the company today.
- vCY represents the value of the company yesterday.
- C1 is a correction parameter set to -206.1103731082200000
- C2 is a correction parameter set to 0.0213265642264200
- C3 is a correction parameter set to 0.9947712774802800

The formula is based on the linear regression explained previously (the parameters C1, C2 and C3 are the same). This regression takes the values that the RSI will take in function of the IBEX35 stock market closing value.

The purpose of this heuristic is that it can be applied to any company contained on the IBEX35. To make that possible, a corrector factor is applied to the RSI value of the day before the day that we are judging, which allows us to obtain a result for a concrete company.

The calculation starts from the RSI of the IBEX35 from the previous day, to subsequently calculate the difference between this RSI and the RSI of the current day. To that difference, a coefficient that indicates the relation between the current tendency of the IBEX35 and the considered action will be applied.

$$Coefficient\ t = \frac{growthRateCompany}{growthRateMarket}$$

In any case, the calculation of the RSI for a concrete action in the current day will always start in the value of the RSI of the IBEX35 on the previous day because it does not save memory.

Optimal RSI with Memory:

The formula of this heuristic is the following (explained in several steps):

$$rsiToday = C1 + (C2 * vMT) + C3$$

$$rsiYdayMarket = C1 + (C2 * vMY) + C3$$

$$restRSI = rsiToday - rsiYdayMarket$$

$$growthRateCompany = (vCT - vCY) / vCY$$

$$growthRateMarket = (vMT - vMY) / vMY$$

$$RSI = rsiYday + (restRSI * \frac{growthRateCompany}{growthRateMarket})$$

Where:

- vMT represents the value of the market today.
- vMY represents the value of the market yesterday.
- vCT represents the value of the company today.
- vCY represents the value of the company yesterday.
- rsiYday represents the value of the RSI of the market yesterday.
- C1 is a correction parameter set to -206.1103731082200000
- C2 is a correction parameter set to 0.0213265642264200
- C3 is a correction parameter set to 0.9947712774802800

If the value rsiYday is empty, the value is calculated with the following formula (when the optimal RSI with memory process starts, the first value of rsiYday will not exist, and for this reason needs to be calculated).

$$rsiYday = C1 + (C2 * vMY) + C3$$

It is the same calculation that was explained in the model without memory with the only difference that, in this case, for the calculi of the RSI of a concrete company on the current day, it will start from the value of the RSI for that concrete company the previous day instead of starting with the IBEX35 RSI.

The reason for this change is because in this mode, as you do not always have to start with the same value (IBEX35 RSI of the day previous to the current day), each company will have an RSI with a tendency not necessarily linked with IBEX5 tendencies.

4. Evaluation

The evaluation of the system consists of two parts:

- In the first place, the evaluation of the neural network and how it is able to predict will be evaluated. More in depth, in this evaluation the best configuration of the neural network setup to calculate better predictions using iRSI is chosen.
- The second part of the evaluation, done in Study 2, consists of querying the neural networks with a certain number of values and checking the signals that the RSI value recommends (buy or sell). The aim is to find out the accuracy of the iRSI.

4.1. STUDY 1: Evaluating generalization of the neural network

The first part of the evaluation of the system consisted of the evaluation of several neural networks with different configurations in order to choose the one that showed better results in terms of investment performance.

4.1.1. Research Design

The aim of this study is to find out which neural network configuration provides better results for the CAST. Different neural networks configuration schemes will be tested and their output will be compared with real values in order to choose the one that achieves better prediction rates. Predictions and neural network configurations will be performed in the scenario of RSI Prediction. The process will be done following the steps as follows:

1. Set up a valid neural networks configuration for RSI calculation.
2. Train neural network setup.
3. Compare results of every set up with a sample of real data.

In what follows a description of these steps is depicted.

1. Set up valid neural networks configuration for the RSI.

The first step is to specify valid neural network configurations. This is done according to these parameters:

- **Complexity:** Represents the complexity of the neural network. Depending on the value, the topology and values will change. The complexity of the neural network has two values (low and medium). Depending on each value, the neural network has a concrete topology.
 - **Low Complexity:** The topology of the neural network with low complexity is shown in Table 2 and all the values (activation function, learning algorithm, etc.) are explained in Section 3.1.
 - **Medium Complexity:** The topology of the neural network with medium complexity has two hidden layers instead of one. Table 4 shows the main configuration values of this topology.

Layer	Input Neurons	Output Neurons	Activation Function
Input	32	165	Laguarre (5 Taps, 1 Tap Delay)
Hidden 1	165	3	TanH + Laguarre (1 Tap/Delay)
Hidden 2	3	1	TanH + Laguarre (1 Tap/Delay)

Output	1	1	Bias
--------	---	---	------

Table 4. Network Topology Values (Medium Complexity)

Momentum values vary from 0 to 1 increasing 0.1 in each change. In the final table of neural networks where the results are shown, only the network with the best momentum value is shown. Figure 4 shows the topology of the network.

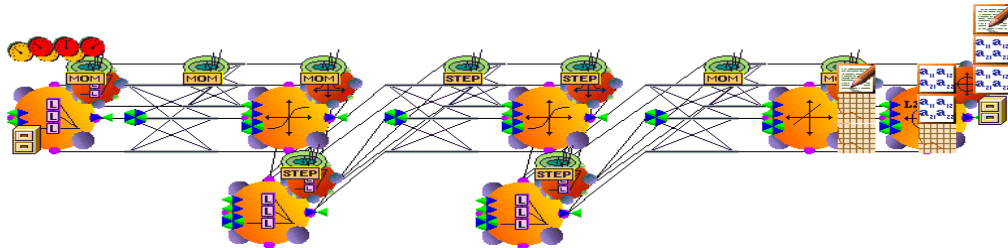


Figure 4. Representation of the network with medium complexity

- **Number of input parameters:** Indicates the number of input parameters (including desired value that is also an input value, but is not reflected in the parameters column) that the network receives.
- **Parameters:** Indicates the parameters given to the neural network. Available parameters are:
 - **AV:** Represents the action value of the market.
 - **RSI[X-Y]:** Represents the RSI values from RSI(X) to RSI(Y) (both included).
 - **RSI(X, ..., Z):** Represents RSI(X), RSI(...), and RSI(Z) values.
 - **OR:** Represents the optimal RSI value.

In the second place, and taking into account configuration parameters, valid neural network configurations are shown in Table 5:

Name	Complexity	# Input Parameters	Parameters
RNAR1	Low	32	AV, RSI[5-35]
RNAR2	Low	2	AV
RNAR3	Low	5	AV, RSI(9,14,30)
RNAR4	Medium	32	AV, RSI[5-35]
RNAR5	Medium	2	AV
RNAR6	Medium	5	AV, RSI(9,14,30)

Table 5. Table of neural networks used to predict the RSI

2. Train neural networks.

Each neural network was trained 10 times to check that the network was working properly. Ten percent of the data from the sample was used for this task.

3. Compare results of every setup with a sample of real data.

The comparison will be made using two measures:

- **Testing OK:** Represents the number (and percentage) of correct results comparing the neural network with real values.
- **M-Testing OK:** Represents the number (and percentage) of correct results including values of confidence for RSI prediction. A value of confidence of five units of margin was used. This value is used because, when calculating the RSI, it can be assumed that there is a little variation in the RSI values in which we can confide in a movement of the value in the way the CAST is predicting.

After obtaining Testing-OK and M-Testing-OK values, the neural network that presents better Testing-OK and M-Testing OK values (higher percentages) will be chosen.

4.1.2. Sample

The sample was composed of a total of 4,061 market prices corresponding to 16 years of stock market prices from the IBEX 35 (Spain). Agreeing with Krogh et al. (1995), 80% of the data was used for training and 20% for cross-validation (812 values).

4.1.3. Results& Discussion

Table 6 shows results from all neural networks tested:

Name	Complexity	# Input Parameters	Parameters	Testing Ok	M-Testing Ok (5 VOC)
RN1	Low	32	AV, RSI[5-35]	7.881%	65.517%
RN2	Low	2	AV	8.867%	75.862%
RN3	Low	5	AV, RSI(9,14,30)	11.084%	80.296%
RN4	Medium	32	AV, RSI[5-35]	3.941%	37.931%
RN5	Medium	2	AV	7.882%	63.054%
RN6	Medium	5	AV, RSI(9,14,30)	7.389%	53.695%

Table 6. Table of results of best neural networks for predicting the RSI

The results show that the RN3 configuration provides the best results from the set analyzed. Using the M-Test Column, the results show that the RN3 configuration can predict 80% (652 cases out of 812) in an accurate way, which is a very good approach. With the objective of verifying whether the results presented statistically significant differences among neural networks configurations, the statistical method analysis of variance (ANOVA) was used to carry out analysis of variance among groups using the SPSS tool. The level of statistical significance was set to 0.05. One-way ANOVA was used in order to test for differences between two or more independent groups (in our case, six groups). The results of the test indicate that groups present significant differences indicated by the statistical value ($F(811)= 89.704$, $p<.05$). This circumstance implies that, from a statistical point of view, there is a difference among predictions. However, it is important to work out whether there is a difference between the best prediction configuration and real values. It was done using the statistical method Student's t-test (comparison of two means). The results of this test showed significant differences between real values and the best predictions corresponding to RN3 ($t(811)=20.716$, $p<.05$). This circumstance reveals that there are ways to improve our solution either by better training or new setup definition, although it presents undeniably good results.

In order to proceed with Study 2, the neural network that will be used as a reference for further studies will be the one named RNAR3 for its remarkable capability of generalization according to its results.

4.2. Study 2: Evaluating behavior of a neural network in a real case

4.2.1. Research Design

In Study 2, once the best neural network configuration (RN3) is chosen in Study 1, it is applied to a real market scenario in a given period of time. The aim of this study is to know whether the iRSI proposed in this paper gives better results than the conventional RSI using 14 days as was suggested by Wilder (1978). In Study 2, this method will be noted by RSI14. Given that the CAST can be applied to two different scenarios, whole market prediction (the IBEX 35, in this case), and single-company prediction (pertaining to the IBEX 35), there will be two main tests. The aim of the first test is to compare the iRSI versus RSI14 for the IBEX 35 stock market. The aim of the second test is to prove that the iRSI can predict single-company values pertaining to the IBEX 35 in a more accurate way compared to RSI14. This comparison performed in Test 2 will be made for the iRSI with and without memory (see 3.4 and 3.5).

The comparison method, which is common to test 1 and test 2, is as follows. Given a concrete day and the IBEX 35 index (test 1) or company (test 2), the index is performed (RSI14 and iRSI). It advises one to either “Sell” or “Buy”. Once the action is performed, the value related to this operation (of selling or buying) is compared to the final value after a period of time (from one day to seven days). If the value is coherent with the prediction (lower if a sell command was sent or higher if a buy command was delivered), then this action increments a success counter for the method proved, RSI14 in this case. The final score will be (success operations) / (total operations).

4.2.2. Sample

The sample used in this test consisted of the data obtained in the period between December 16th, 2005 and October 27th, 2009, a total of 812 values. For Test 1, IBEX 35 values were used. For Test 2, fifteen companies in the Spanish stock market (IBEX35) were used, namely, Enagas, ACS, Inditex, Telecinco, Santander, Indra, Abengoa, Iberia, Iberdrola, Repsol, Sacyr, BBVA, Banesto, Telefónica, and Abertis, thus using 812 values per company, a total of 12,180 values.

4.2.3. Results& Discussion

TEST 1

Table 7 shows the results of the chosen neural network for iRSI prediction compared with RSI14 predicting IBEX 35 behavior:

Period	RSI14	iRSI
1 Day	45.65%	53.83%
2 Days	43.00%	55.46%
3 Days	43.48%	56.15%
4 Days	42.27%	58.47%

5 Days	41.79%	58.61%
6 Days	41.79%	58.61%
7 Days	42.27%	58.20%

Table 7. Table of prediction results using RSI

The results show a better performance of the iRSI compared to RSI14. All measures present higher values in the iRSI and all of them are above 50%. On the other hand, none of the measures by RSI14 present measures above 46%. Higher success values for the iRSI are present in 5 and 6-days periods (58.61%). The first test performed in order to find out whether there are significant differences between RSI14 and the iRSI was a Student's t-test (comparison of two means). The test was performed for the predictions made without taking periods into account. The results of this test showed significant differences between the iRSI and RSI14 ($t(11367)=15.407$, $p<.05$). This circumstance, which can be easily inferred from the results, reveals that the iRSI is a better guide for investors in order to predict the IBEX 35 stock market. Applying this same test to pairs of predictions, results show a better prediction as well as a significant difference between every RSI14 and iRSI pair in every case.

The second issue is to find out whether there are significant differences among periods as a whole in the iRSI. To do so, One-way ANOVA was used in order to test for differences between two or more independent groups (in this case, seven groups). The result of this test pointed out that there are no significant differences among periods for the iRSI ($F(5683)= 1.236$, $p>.05$). This asseveration can be inferred also from the results, showing just slight differences among scores.

The application of neural networks to Ibex35 stock prediction is not new (e.g., Fernández-Rodríguez, González-Martel, & Sosvilla-Rivero, 2000; Mingo López, Díaz, Palencia, Santos, & Jiménez, 2002; Pérez-Rodríguez, Torra&Andrada-Félix, 2005a; Pérez-Rodríguez, Torra&Andrada-Félix, 2005b). In these cases, neural networks provide a reasonable description of asset price movements. In our case, in which the aim is to improve the RSI, the results show an unquestionable empowerment of results compared to RSI14.

TEST 2

Finally, we show the results of the prediction using the neural network (RNAR3) adapted with the heuristic of RSI calculation with and without memory using some companies from the IBEX35 market. Table 8 shows the results of the application of RSI14 for selected enterprises pertaining to the IBEX 35 stock market:

Company	RSI14 PR 1D	RSI14 PR 2D	RSI14 PR 3D	RSI14 PR 4D	RSI14 PR 5D	RSI14 PR 6D	RSI14 PR 7D
Enagas	52.36%	56.22%	54.51%	56.22%	57.51%	57.94%	55.36%
ACS	43.60%	46.71%	44.29%	41.18%	41.52%	41.18%	42.21%
Inditex	49.78%	48.89%	49.33%	49.78%	48.00%	45.33%	43.11%
Telecinco	44.30%	44.73%	45.15%	45.15%	40.93%	41.77%	41.77%
Santander	49.57%	50.00%	53.02%	55.60%	53.02%	50.43%	50.43%
Indra	58.38%	59.46%	62.70%	63.78%	65.95%	65.41%	65.41%
Abengoa	52.40%	50.92%	48.71%	50.18%	47.97%	49.45%	51.29%

Iberia	48.28%	49.43%	42.53%	43.30%	38.70%	41.38%	38.31%
Iberdrola	49.29%	46.79%	46.79%	49.29%	47.14%	44.64%	45.71%
Repsol	48.24%	51.37%	52.16%	50.98%	50.20%	53.33%	52.55%
Sacyr	48.37%	46.88%	49.85%	48.66%	45.10%	44.81%	43.03%
BBVA	47.58%	50.40%	50.81%	52.02%	49.60%	51.21%	50.00%
Banesto	42.97%	46.39%	45.25%	42.21%	41.06%	38.02%	39.92%
Telefónica	49.09%	48.18%	47.73%	45.91%	46.82%	46.82%	46.82%
Abertis	55.90%	52.31%	52.31%	49.74%	50.26%	49.74%	47.69%

Table 8. Table of prediction results in companies using RSI14

Table 9 shows results of the application of the iRSI without memory using RNAR3. Figures in bold letters means better results for the iRSI without memory than RSI14.

Company	iRSIPR 1D	iRSIPR 2D	iRSIPR 3D	iRSIPR 4D	iRSIPR 5D	iRSIPR 6D	iRSIPR 7D
Enagas	46.49%	48.43%	49.93%	48.43%	48.58%	48.43%	47.83%
ACS	47.31%	50.75%	49.85%	50.90%	51.79%	51.49%	51.19%
Inditex	50.30%	52.99%	54.34%	54.94%	53.29%	52.10%	54.57%
Telecinco	49.41%	54.45%	55.49%	56.68%	56.97%	56.82%	56.23%
Santander	51.49%	52.99%	54.48%	55.52%	55.37%	55.52%	55.97%
Indra	48.49%	53.02%	54.38%	55.89%	56.19%	55.44%	55.29%
Abengoa	53.93%	53.78%	52.15%	52.59%	53.33%	52.15%	52.74%
Iberia	49.93%	49.93%	48.14%	50.07%	49.48%	51.12%	51.56%
Iberdrola	53.06%	52.61%	53.50%	53.50%	54.25%	52.61%	53.35%
Repsol	56.54%	53.83%	50.38%	52.33%	52.78%	51.43%	52.48%
Sacyr	55.44%	55.14%	55.14%	56.80%	57.40%	57.85%	59.67%
BBVA	54.40%	52.91%	54.99%	54.99%	54.55%	54.55%	54.40%
Banesto	52.02%	53.36%	53.66%	54.41%	55.46%	54.26%	55.31%
Telefónica	48.80%	49.10%	49.40%	48.05%	47.46%	47.46%	47.90%
Abertis	52.82%	51.48%	51.63%	53.41%	53.56%	54.45%	54.38%

Table 9. Table of prediction results in companies using the iRSI without memory

Table 10 shows results of the application of the iRSI with memory using RNAR3. Figures in bold letters mean better results for the iRSI without memory than RSI14.

Company	iRSIPR 1D	iRSIPR 2D	iRSIPR 3D	iRSIPR 4D	iRSIPR 5D	iRSIPR 6D	iRSIPR 7D
Enagas	50.39%	52.70%	54.44%	55.02%	55.98%	55.41%	56.37%
ACS	47.03%	51.40%	52.62%	52.62%	52.27%	52.27%	54.02%
Inditex	49.65%	50.53%	51.41%	51.94%	51.94%	50.88%	52.30%
Telecinco	45.73%	46.68%	46.99%	48.73%	47.78%	50.63%	49.92%
Santander	50.29%	50.44%	51.32%	52.20%	52.05%	54.11%	54.55%
Indra	48.55%	51.06%	52.22%	54.34%	56.65%	55.30%	55.11%
Abengoa	51.04%	49.87%	48.04%	49.74%	50.26%	49.48%	50.72%
Iberia	47.30%	47.67%	47.42%	48.77%	48.77%	50.49%	51.91%
Iberdrola	49.41%	48.50%	49.15%	48.37%	49.15%	46.68%	47.72%
Repsol	56.27%	54.53%	54.00%	55.33%	57.33%	56.00%	57.47%

Sacyr	49.14%	48.78%	48.78%	50.00%	49.88%	49.27%	50.98%
BBVA	54.45%	53.82%	56.01%	56.47%	56.79%	57.57%	57.57%
Banesto	51.36%	50.15%	51.97%	52.27%	53.18%	52.12%	54.55%
Telefónica	48.55%	48.06%	48.87%	48.39%	48.23%	48.06%	49.19%
Abertis	50.86%	49.88%	49.75%	51.48%	52.71%	54.06%	53.69%

Table 10 – Table of prediction results in companies using the iRSI with memory

As can be observed in the result section of the evaluation, the use of neural networks using heuristic formulas calculated by linear regression of financial factors for training can improve RSI14. Tables 9 and 10 show better prediction rates in the iRSI without memory; it presents worse values than the iRSI with memory in two cases, always in the six and seven-day prediction. Having in mind that technical analysis is indicated in short-term predictions, it is feasible to assume that broader periods can make worse predictions. Thus, the method selected will be the iRSI without memory. This method brings better results than RSI14 in at least 73.3% of the cases.

The global accuracy of predictions is quite similar for the three methods studied (RSI14= 48.92%, iRSI without memory = 52.76% and iRSI with memory = 51.47%). The highest success prediction rates correspond to 2D in the case of RSI14 (49.91%) and 7D in the case of iRSI without memory (53.52%) and iRSI with memory (53.07%). The results are very close and none of them present statistically significant differences. However, it is important to point out the difference in better predictions within the range: 7D for iRSI and 2D for RSI14. This could mean that the conventional RSI14 predicts the shortest ranges and iRSI ranges above 3 days in a better way (from 3 days on, the results are practically the same).

According to enterprise predictions on average, the best prediction results as a whole came from INDRA (63.01%) and RSI14 and the worst from BANESTO and RSI14 (42.26%), and taking isolated values ranked INDRA in RSI14 as the best (65.95%) for the 5D period and BANESTO as the worst (38.02%) for the 6D prediction. These results show a more predictive behavior for the iRSI, having lower standard deviation values (0.024 for the iRSI without memory and 0.026 for the iRSI with memory) than RSI14 (0.055), thus, being more applicable to a wider range of values.

There are two enterprises, namely, INDRA and ENAGAS, in which iRSI with and without memory predictions are, in all cases, worse than RSI14. The authors failed to find a rational reason why. The important issue here is that both INDRA and ENAGAS have very accurate predictions in RSI14. The most feasible explanation is the chaotic data nature label given in the literature (e.g., Wen et al., 2010). However, the authors rely on iRSI development, and as future work suggest expanding the sample in order to investigate this phenomenon.

The main conclusion to this Test 2 is that the iRSI without memory is more predictable (lower standard deviation) and also more accurate (higher percentages) than the other methods. This is done in a better way using 7D as the time frame. However, it is important to say that several values are more accurately predicted using RSI14, possibly due to the stochastic nature of the stock markets.

5. Conclusions and Future Work

The current paper describes a research project about the generation of RSI values to create systems capable of generating automated or semi-automated investments in certain companies in the Spanish IBEX35 stock market. In this paper, the main work is based on the study case of generating a heuristic for a concrete market and applying some corrections factor in order to be able to generate good investment results for concrete companies of the sector. The generation of the RSI, known as the iRSI, also was tested using two variants in its calculation: using memory and not using it. This variant also implies the modification of the main heuristic formula created to calculate optimal RSI values. Both approaches were generated, studied and evaluated and the final conclusion reached was that RSI optimal generation (without memory) is the better option.

This paper was based only on the RSI financial indicator and the heuristic methods applied where generated to create a single heuristic formula for the IBEX35 stock market. The current paper proposes four types of initiatives which should be explored in future research. In the first place, our future work plans to generate a heuristic for each company by analyzing its data. In the second place, we will extend the application of the iRSI to a broader sample: more companies pertaining to the IBEX 35, more indexes both national and international and, of course, as stated before, a bigger time frame. In the third place, we will tune the iRSI to adapt it to momentum in the market (upward or downward trend). Finally, we will expand the research to investigate broader technical analysis indexes like the MACD (Moving Average Convergence / Divergence) financial indicator or Bollinger Bands.

Acknowledgements

This work is supported by the Spanish Ministry of Industry, Tourism, and Commerce under the EUREKA project SITIO (TSI-020400-2009-148), SONAR2 (TSI-020100-2008-665), INNOVA 3.0 (TSI-020100-2009-612) and GO2 (TSI-020400-2009-127).

References

- Abhyankar, A., Copeland, L. S., & Wong, W. (1997). Uncovering nonlinear structure in real-time stock-market indexes: the S&P 500, the DAX, the Nikkei 225, and the FTSE-100. *Journal of Business and Economic Statistics*, 15 (1), 1–14.
- Aiken, M., & Bsat, M. (1994). Forecasting market trends with neural networks. *Information Systems Management*, 6(4), 42–48.
- Armano, G., Marchesi, M., & Murru, A. (2005). A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences*, 170 (1), 3–33.
- Armano, G., Murru, A., & Roli, F. (2002). Stock market prediction by a mixture of genetic-neural experts. *International Journal of Pattern Recognition and Artificial Intelligence*, 16 (5), 501–526.
- Arulampalam, G., & Bouzerdoum, A. (2003). A generalized feedforward neural network architecture for classification and regression. *Neural Networks*, 16 (5-6), 561-568.

- Austin, M., & Looney, C. (1997). Security market timing using neural network models. *New Review of Applied Expert Systems*, 3, 3–14.
- Bao, D., & Yang, Z. (2008). Intelligent stock trading system by turning point confirming and probabilistic reasoning. *Expert Systems with Applications*, 34 (1), 620–627.
- Black, A. J., & Mcmillan, D. G. (2004). Non-linear predictability of value and growth stocks and economic activity. *Journal of Business Finance and Accounting*, 31 (3/4), 439–474.
- Brownstone, D. (1996). Using percentage accuracy to measure neural network predictions in stock market movements. *Neurocomputing*, 10, 237–250.
- Chang, P.C., & Liu, C.H. (2008). A TSK type fuzzy rule based system for stock price prediction. *Expert Systems with Applications*, 34 (1), 135–144.
- Chang, P.C., Liu, C.H., Lin, J.L., Fan, C.Y., & Ng, C.S.P. (2009). A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications*, 36 (3), 6889–6898.
- Chang, P.C., Wang, Y. W., & Liu, C. H. (2007). The development of a weighted evolving fuzzy neural network for PCB sales forecasting. *Expert Systems with Applications*, 32(1), 86–96.
- Chavarnakul, T., & Enke, D. (2008). Intelligent technical analysis based equivolume charting for stock trading using neural networks. *Expert Systems with Applications*, 34 (2), 1004–1017.
- Chen, A.S., Leung, M.T., & Daouk, H. (2003). Application of neural networks to an emerging financial market: Forecasting and trading the Taiwan stock index. *Computers and Operations Research*, 30 (6), 901–923.
- Chen, A.S., & Leung, M.T. (2004). Regression neural network for error correction in foreign exchange forecasting and trading. *Computers and Operations Research*, 31 (7), 1049–1068.
- Chen, J.S., Hou, J.L., Wu, S.M., & Chang-Chien, Y.W. (2009). Constructing investment strategy portfolios by combination genetic algorithms. *Expert Systems with Applications*, 36 (2) Part 2, 3824–3828
- Chen, Y., Mabu, S., Shimada, K., & Hirasawa, K. (2009). A genetic network programming with learning approach for enhanced stock trading model. *Expert Systems with Applications*, 36 (10), 12537–12546.
- Chenoweth, T., Obradovic, Z., & Stephenlee, S. (1996). Embedding technical analysis into neural network based trading systems. *Applied Artificial Intelligence*, 10(6), 523–541.
- Chiam, S.C., Tan, K.C., & Al Mamun, A. (2009). Investigating technical trading strategy via an multi-objective evolutionary platform. *Expert Systems with Applications*, 36 (7), 10408–10423.
- Chiu, D.Y. & Chen, P.J. (2009). Dynamically exploring internal mechanism of stock market by fuzzy-based support vector machines with high dimension input space and genetic algorithm. *Expert Systems with Applications*, 36 (4), 1240–1248.

- Cybenko, G. (1989). Continuous valued neural networks with two hidden layers are sufficient. *Mathematics of Control, Signal and Systems*, 2, 304-314.
- Edwards, R., & Magee, J. (1997). *Technical analysis of stock trends* (7th edition). New York: Amacom.
- Enke, D., & Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 29(4), 927–940.
- Fernández-Rodríguez, F., González-Martel, C. & Sosvilla-Rivero, S. (2000). On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market. *Economics Letters*, 69 (1), 89-94.
- Fine, T.L. (1999). *Feedforward neural network methodology*. New York: Springer.
- Grudnitski, G., & Osburn, L. (1993). Forecasting S&P and gold futures prices: an application of neural networks. *Journal of Futures Market*, 13 (6), 631–643.
- Hassan, M.R., Nath, B., & Kirley, M. (2007). A fusion model of HMM, ANN and GA for stock market forecasting. *Expert Systems with Applications*, 33 (1), 171–180.
- Hiemstra, C., & Jones, D. (1994). Testing for Linear and Nonlinear Granger Causality in the Stock Price-volume Relation. *Journal of Finance*, 49(5), 1639–1664.
- Huang, W., Nakamori, Y., & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine source. *Computers and Operations Research*, 32(10), 2513–2522.
- Huang, K., & Yu, H.K. (2005). A Type 2 fuzzy time series model for stock index forecasting. *Physica A: Statistical Mechanics and its Applications*, 353, 445–462.
- Kim, K.J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55 (1/2), 307–319.
- Kim, K.J. (2004). Toward Global Optimization of Case-Based Reasoning Systems for Financial Forecasting. *Applied Intelligence*, 21 (3), 239–249.
- Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural network. *Proceedings of the International Joint Conference on Neural Networks* (pp. 1–6). San Diego, CA.
- Kovalerchuk, B., & Vityaev, E. (2000). *Data mining in finance: advances in relational and hybrid methods*. Kluwer Academic.
- Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, 7, 231-238.
- Kwon, Y.K., & Moon, B.R. (2007). A hybrid neurogenetic approach for stock forecasting. *IEEE Transactions on Neural Networks*, 18(3), 851-864.

- Lai, R.K., Fan, C.Y., Huang, W.H., & Chang, P.C. (2009). Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Systems with Applications*, 36 (2), 3761-3773.
- Lam, M. (2004). Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision Support Systems*, 37 (4), 567–581.
- Lee, T.S., & Chen, N.J. (2002). Investigating the information content of non-cash-trading index futures using neural networks. *Expert Systems with Applications*, 22 (3), 225–234.
- Lee, T.S., & Chiu, C.C. (2002). Neural network forecasting of an opening cash price index. *International Journal of Systems Science*, 33 (3), 229–237.
- Leigh, W., Hightower, R., & Modani, N. (2005). Forecasting the New York stock exchange composite index with past price and interest rate on condition of volume spike. *Expert Systems with Applications*, 28(1), 1–8.
- Leigh, W., Modani, N., Purvis, R., & Roberts, T. (2002). Stock market trading rule discovery using technical charting heuristics. *Expert Systems with Applications*, 23 (2), 155-159.
- Leigh, W., Purvis, R., & Ragusa, J.M. (2002). Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. *Decision Support Systems*, 32 (4), 361–377.
- Lo, A., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithm, statistical inference, and empirical implementation. *Journal of Finance*, 55 (4), 1705–1765.
- Lu, C.J., Lee, T.S., & Chiu, C.C. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, 47 (2), 115–125.
- Majhi, R., Panda, G., & Sahoo, G. (2009). Development and performance evaluation of FLANN based model for forecasting of stock markets. *Expert Systems with Applications*, 36 (3) Part 2, 6800-6808.
- Malkiel, B. G. (1995). *A random walk down wall street*. New York, NY: Norton & Co.
- Mingo López, L. F., Díaz, M. A., Palencia, V., Santos, E., & Jiménez, P. (2002). IBEX-35 stock market forecasting using time delay connections in enhanced neural networks. In *World multiconference on systemics, cybernetics and informatics* (pp. 455–460).
- Murata, N., Yoshizawa, S., & Amari, S. (1994). Determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 5 (6), 865-872.
- Murphy, J. J. (1999). *Technical analysis of the financial markets*. Paramus, NJ: New York Institute of Finance.
- Oh, K.J., & Kim, K.J. (2002). Analyzing stock market tick data using piecewise non linear model. *Expert System with Applications*, 22 (3), 249–255.

- Oh, K.J., Kim, T.Y., & Min, S. (2005). Using genetic algorithm to support portfolio optimization for index fund management. *Expert Systems with Applications*, 28 (2), 371–379.
- Oh, K.J., Kim, T.Y., Min, S.H., & Lee, H.Y. (2006). Portfolio algorithm based on portfolio beta using genetic algorithm. *Expert Systems with Applications*, 30 (3), 527–534.
- Pai, P.F., & Lin, C.S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*, 33 (6), 497–505.
- Pérez-Rodríguez, J. V., Torra, S., & Andrada-Félix, J. (2005). Are Spanish Ibex35 stockfuture index returns forecasted with non-linear models? *Applied Financial Economics*, 15(14), 963-975.
- Pérez-Rodríguez, J. V., Torra, S., & Andrada-Félix, J. (2005). STAR and ANN models: forecasting performance on the Spanish “Ibex-35” stock index. *Journal of Empirical Finance*, 12 (2005), 490–509.
- Peters, E.E. (1994). *Fractal market analysis: applying chaos theory to investment and economics*. New York: Willey & Sons, Inc.
- Picard, R., & Cook, D. (1984). Cross-validation of regression models. *Journal of the American Statistical Association*, 79 (387), 575–583
- Potvin, J.Y., Soriano, P., & Vallee, M. (2004). Generating trading rules on the stock markets with genetic programming. *Computers & Operations Research*, 31 (7), 1033–1047.
- Pring, M.J. (1991). *Technical analysis explained*. New York: McGraw-Hill.
- Roh, T.H. (2007). Forecasting the volatility of stock price index. *Expert Systems with Applications*, 33 (4), 916–922.
- Rumelhart, D.E., & McClelland, J.L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*. Volume I. Cambridge, MA: MIT Press.
- Saad, W.E., Prokhorov, V.E., & Wunsch, C.D. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks*, 9(6), 1456–1470.
- Schoeneburg, E. (1990). Stock price prediction using neural networks: a project report. *Neurocomputing*, 2(1), 17–27.
- Tan, A., Quek, C. & Yow, K.C. (2008). Maximizing winning trades using a novel RSPOP fuzzy neural network intelligent stock trading system. *Applied Intelligence*, 29(2), 116-128.
- Thammano, A. (1999). Neuro-fuzzy Model for Stock Market Prediction. In Dagli, C.H., A.L. Buczak, J. Ghosh, M.J. Embrechts, & O. Ersoy (Eds.) *Smart Engineering System Design: neural networks, fuzzy logic, evolutionary programming, data mining, and complex systems*. In *Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE '99)*. New York: ASME Press, 587–591.

- Thawornwong, S., & Enke, D. (2004). The adaptive selection of financial and economic variables for use with artificial neural networks. *Neurocomputing*, 56, 205–232.
- Trippi, R.R., & DeSieno, D. (1992). Trading equity index futures with a neural network. *Journal of Portfolio Management*, 19 (1), 27–33.
- Vanstone, B., & Tan, C.N.W. (2005). Artificial neural networks in financial trading, In M. Khosrow-Pour (Ed.), *Encyclopedia of information science and technology*. Idea Group.
- Wang, J.L., & Chan, S.H. (2006). Stock market trading rule discovery using two-layer bias decision tree. *Expert Systems with Applications*, 30 (4), 605-611.
- Wang, Y.F. (2003). Mining stock prices using fuzzy rough set system. *Expert System with Applications*, 24 (1), 13–23.
- Wen, Q., Yang, Z., Song, Y., & Jia, P. (2010). Automatic stock decision support system based on box theory and SVM algorithm. *Expert Systems with Applications*, 37(2), 1015-1022.
- White, H. (1988). Economic prediction using neural networks: The case of IBM daily stock returns, In *Proceedings of the 2nd annual IEEE conference on neural networks*, II (pp. 451–458).
- Wilder, J. W. Jr (1978). *New Concepts in Technical Trading Systems*. Greensboro, NC: Hunter Publishing Company.
- Yao, J., & Herbert, J.P. (2009). Financial time-series analysis with rough sets. *Applied Soft Computing*, 9 (3), 1000–1007.
- Yao, J., Li, Y., & Tan, C.L. (2000). Option price forecasting using neural networks. *Omega*, 28 (4), 455–466.
- Yao, J., Tan, L. C., & Poh, H.L. (1999). Neural networks for technical analysis: a study on KLCI. *International Journal of Theoretical and Applied Finance*, 2(2), 221–241.
- Yoon, Y., & Swales, J. (1991). Prediction stock price performance: A neural network approach. In *Proceeding of 24th annual Hawaii international conference on system science* (pp. 156–162).
- Yu, X., & Chen, G. (1996). Efficient backpropagation learning using optimal learning rate and momentum. *Neural Networks* 10(3), 517-527.