European Journal of Economic and Social Systems **14** N° 1 (2000) 81-91

Non-linear financial time series forecasting – Application to the Bel 20 stock market index

A. LENDASSE[1], E. DE BODT[2], V. WERTZ[1] AND M. VERLEYSEN[3]

Abstract. – We developed in this paper a method to predict time series with non-linear tools. The specificity of the method is to use as much information as possible as input to the model (many past values of the series, many exogenous variables), to compress this information (by a non-linear method) in order to obtain a state vector of limited size, facilitating the subsequent regression and the generalization ability of the forecasting algorithm and to fit a non-linear regressor (here a RBF neural network) on the reduced vectors. We show that this method is able to find non-linear relationships in artificial and real-world financial series. On a difficult task, which consists in forecasting the tendency of the Bel 20 stock market index, we show that this method improves the results compared both to linear models and to non-linear ones where the non-linear compression is not used.

1. Introduction

1.1. *Financial analysis background*

Forecasting a time series is a common problem in many domains of science (electricity, hydrology, etc.), and has been addressed for a long time by statisticians. Predicting a financial series, as a stock market index or an exchange rate, remains however a very specific task.

The study of the behaviour of stock market prices has begun since a long time now in finance. Already in 1965, Fama (1965) clearly put to light the highly stochastic nature of their behaviour.

It seems that Bachelier (1914) was the first to propose the theory of random walk to characterise the changes of security prices through time. Fama (1965) analyses the distri-

[1] Université catholique de Louvain, CESAME-AUTO, 4 av. G. Lemaître, 1348 Louvain-la-Neuve, Belgium, {lendasse, wertz}@auto.ucl.ac.be

[2] Université de Lille 2, ESA-GERME Université catholique de Louvain, IAG-FIN, Belgium, debodt@fin.ucl.ac.be

[3] Université catholique de Louvain, CERTI, 3 pl. du Levant, 1348 Louvain-la-Neuve, Belgium, verleysen@dice.ucl.ac.be. Michel Verleysen is a research associate of the belgian FNRS.

Keywords: time series forecasting, neural networks stock index prediction, curvilinear component analysis.

bution on a large (at least at this time) data set (the thirty stocks of the Dow-Jones Industrial Average during the period 1957-1962). He shows that empirical evidence seems to confirm the random walk hypothesis: a series of price changes has no memory ("the past can not be used to predict the future in any meaningful way [1]"). The main theoretical explanation that lies behind this observation is the efficient market hypothesis (EMH). While a detailed explanation of this concept is beyond the scope of this paper, the main ideas are the following. If a statistically significant serial dependence exists within time series of financial security prices, the community of financial analysts will immediately exploit it. Security price changes can therefore be only explained by the arrival of new information, which, by definition, can not be forecasted. The EMH has received a lot of empirical support in the academic literature during the seventies and the eighties. This line of thought has always been received with a lot of scepticism, not to say some irony, in the professional community, which led to the use of charts and technical analysis rules [2]. Professionals have always claimed that classical statistical tests are mainly linear and therefore, unable to capture the complex patterns the price changes exhibit. But things seem to have changed (at least partially) during these last years. As stated by Campbell *et al.* (1967), "several authors signal a growing interest in technical analysis among financial academics, and so it may become a more active research area in the near future". The work that we propose here can be viewed as a contribution to this field of research. We apply non-linear statistical tools (we hope in a clever way) to see they can "break" the random walk hypothesis.

1.2. *Time series analysis*

The succession of values in a time series is usually influenced by some external (or exogenous) information. If this information is not known, only the past values of the series itself can be used to build a *model*, *i.e.* a mathematical function of the form

$$x_{t+1} = f_\theta(x_t, x_{t-1}, \ldots, x_{t-N+1}) . \quad (1)$$

where an unknown new value x_{t+1} is estimated from the know current and past values of x (Ljung, 1997; Weigend *et al.*, 1994). The parameters θ of the model f_θ are chosen according to the information available, *i.e.* to all known values of x; this step if called *learning* or *fitting*.

Sometimes other information is available (for example the outside temperature when one tries to estimate the electricity consumption in a country, or foreign stock market indices when one tries to estimate one of these indices). In this case, it is a good idea to use this external information in the model, usually in the form

$$x_{t+1} = f_\theta(x_t, x_{t-1}, \ldots, x_{t-N+1}, y_t^1, y_t^2, \ldots, y_t^P) \quad (2)$$

where the values at time t of P external (*exogenous*) variables are used in the model.

---

1. We have to mention that the main focus of the author is to test if the security price changes have a finite variance or not.

2. Technical analysis is defined par Edward and Magee has "the science of recording, usually in graphic form, the actual history of trading (price changes, volume of transactions…) in a certain stock or in "the averages" and then deducting from that pictured history the probable future trend".

Most widely known prediction tools use *linear* models f_θ (Box and Jekins, 1976). Artificial neural networks now offer an interesting alternative: the use of *non-linear* models (Sjoberg *et al.*, 1995). Non-linear models are by definition more powerful, since they give more possibilities in the choice of the input-output relation; of course non-linear models include linear ones. Working with non-linear models is however more difficult: the increased possibilities may be seen as supplementary degrees of freedom, leading to a better fitting of the model to the known values, but to a worst generalization ability of the model on unknown data. This *learning-generalization* dilemma, similar to the *bias-variance* dilemma in statistics, is the main limitation of artificial neural networks (ANNs). Indeed some ANNs have the *universal approximation* property: under mild conditions on the data, they can fit any data set with an arbitrary high precision, provided that there is a sufficient number of parameters in the model f_θ. However, when there are too many parameters (compared to the number of data available), the *overfitting* phenomenon appears. The known data (used for learning) are well fitted, but the function f_θ has no sense *between* points used for learning. Let us imagine how three points could be *fitted* by a 10-order polynomial…

This overfitting problem increases with the model complexity, and is thus more difficult to handle when many input variables (past values and exogenous information) are used. In this paper, we show how to cope with this problem: we first choose a large number of variables, containing as much information as possible, and then reduce this number by mathematical tools: estimation of intrinsic dimension and non-linear projection. We illustrate the method on the prediction of the Bel 20 stock market index.

2. Time series forecasting

2.1. *Non-linear regression*

According to equation (2), forecasting a time series is equivalent to choosing a model f_θ and fitting its parameters to the data available (function approximation). In the financial example given below, we use a non-linear model f known as Radial Basis Function (RBF). A RBF is a model of the form

$$f_\theta(x) = \sum_{i=1}^{M} \lambda_i e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \tag{3}$$

where x is the input vector including past values and exogenous variables, and $\theta = \{\mu_i, \sigma_i, \lambda_i\}$ is the parameter set of the model (respectively the center, width and weight of Gaussian functions). Under mild assumptions, RBFs can approximate any data set provided that the number of Gaussian kernels M is sufficiently high (universal approximation property) (Broomhead and Lowe, 1988). Another ANN model, the Multi-Layer Perceptron (MLP) (Werbos, 1974; Rumelhart *et al.*, 1986), has the same property, and is probably more widely used. We use RBFs in our work because the learning process (fitting the parameters of the model) is computationally easier than with MLPs. The learning algorithm used for this work is described in (Verleysen and Hlaváčková, 1994).

It is based on an unsupervised clustering of data points x to choose the centers μ_i and the widths σ_i, and on a supervised fitting of weights λ_i to minimize the mean square error between known output values and estimated ones. Fitting the three sets of parameters independently greatly facilitates training (the only supervised step is linear, unlike learning in a MLP).

2.2. *Input vector*

Past values of the series and exogenous variables are used as information for the learning process. In a real world application, it is difficult to know how much information (in terms of number of variables, or size of input vector x) must be used to properly learn the dynamics of a time series. Obviously, the quantity of information increases with the number of variables. However, we also know that more input variables will lead to more parameters in the function f_θ, which increases the overfitting problem. The main idea of this work is then the following. We first choose a large number of variables which need to be taken into account: sufficiently (probably too many) past values of the series, and many exogenous variables that *could* influence the series. Our purpose will be to transform this set of variables into another smaller set of *state* variables, keeping as much as possible the information contained in the original set. If most of the information is kept, little will be lost during the learning process. Furthermore if the number of state variables is sufficiently lower than the number of initial variables, we will increase the generalization property of the model.

Three steps are thus necessary before fitting the model f_θ:

1. select an original set of R variables;
2. estimate the minimum number S of state variables that could keep most of the information contained in the original ones;
3. transform the initial set of R variables into a set of S state variables.

Step 1 fully depends on the series itself, and on the experience of the user, as we will see in the examples below. Steps 2 and 3 are separated because the dimension S of a possible state space depends on the initial data, and not on *how* (by which method or algorithm) they could be transformed into another set.

The procedure described above is somewhat standard in statistical data analysis. In particular, PCA (Principal Component Analysis) is widely used to reduce the number of variables, with the same goal. Unfortunately, PCA is a purely linear projection method that is unable to catch non-linear relations between data. For example, 2-dimensional data points forming a circle cannot be projected on a 1-dimensional axis (a line) without "flattening" the circle, *i.e.* projecting points from the two half-circles on the same coordinates on the axis. This is a strong drawback that is particularly restrictive when dealing with highly non-linear data as in financial time series. In this work we adopted a method that can be viewed as a non-linear projection.

2.3. *Intrinsic dimension*

One of the key points of the method is to estimate the number S of state variables that will be used for a good prediction. We start with a large number of past values of the

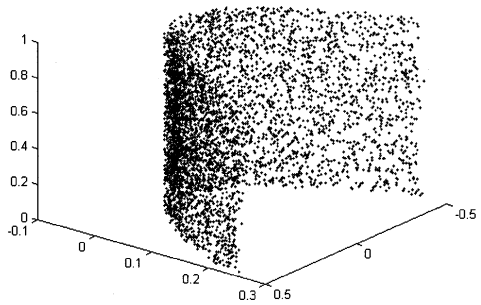

Fig. 1. *Horseshoe distribution.*

series. Together with the exogenous variables, they form the initial *regressor.* The size of this vector is the *regressor order* of the model. In order to determine the optimal regressor order, we will use the notion of *intrinsic* dimension of a set of points. Without going into mathematical details, the intrinsic dimension of a data set can be defined as the minimum number of coordinates that would be necessary to describe the data without loss of information, if these coordinates were measured on curved axes. For example the intrinsic dimension of a set of points forming a string in dimension 2 (or higher) is 1, and the intrinsic dimension of a set of points forming a non-planar surface in dimension 3 (like the well-known horseshoe distribution - see Fig. 1) is 2.

First we build a regressor of size $R = N+P$ with the N last past values of the raw time series and the P exogenous variables (see Eq. (2)). This vector will have to be sufficiently large (both in N and P) to contain all information necessary to a good prediction. One possible solution is to take the optimal regressor for an ARX linear model (Ljung, 1987). Indeed this one is built in such a way that it contains “sufficient” information when used with a linear prediction method, and will thus contain enough information too when used with a potentially more powerful non-linear prediction method. A larger vector can be taken for more security, but will make the rest of the work more difficult. Such a regressor is built for each known time step; the vectors are laid out as rows in a matrix called regressor matrix (or Hankel matrix when there are no exogenous variables).

Since it is supposed that there is an excess of information in the regressor, we will try to reduce its dimension. Step 2 in the above scheme consists in estimating an optimal reduced dimension (the number S of state variables), which will be identified to the intrinsic dimension of the set of points (the regressors) in a $N+P$ -dimensional space. This value will be further referred as the intrinsic dimension of the regressor matrix. It can be interpreted as the number of "non-linearly independent" columns of this matrix: there is a non-linear transformation that makes it possible to rebuild entirely the initial matrix from S columns.

To estimate the intrinsic dimension of the regressor matrix, we use the Grassberger and Procaccia method (Grassberger and Procaccia, 1983); many other methods can however be used to estimate an intrinsic dimension (Takens, 1983; Theiler, 1990; Alligood *et al.*, 1997). It must be mentioned that the concept itself of non-linear dependency is difficult to define. Therefore the intrinsic dimension found by these methods can vary; in difficult

situations, it may be worthwhile to use several methods in order to assess their results. The intrinsic dimension can be a non-integer value; in the following, we will use the integer value nearest to the intrinsic dimension as an approximation of the regressor order defined above.

2.4. *State vector*

The next step consists in building a reduced state vector of size S from each of the original $N+P$ -dimensional regressors.

The set of points defined by the rows of the original regressor matrix form a S-surface in a $N+P$ -dimensional space. If we unfold this S-surface by projecting the $N+P$ -dimensional space onto a S-dimensional one, keeping the topology of the initial set, we obtain a S-dimensional regressor matrix that can be used for further prediction.

Many non-linear "projection" methods exist. Kohonen's self-organising map is probably the most widely known example (Kohonen, 1995). Yet in our experiments we will use another method, the Curvilinear Component Analysis (CCA) (Demartines and Hérault, 1997); unlike the Kohonen maps, this method does not make any assumption on the shape of the projection space, and was found to give better results in our application.

The reduced S-dimensional state vector is then used as input vector x in the RBF model defined by equation (3).

3. An artificial example

In order to test the above method, we build an artificial time series from the non-linear equation

$$x_{t+1} = ax_t^2 + bx_{t-2} + \varepsilon_t. \tag{4}$$

Obviously, the non-linear regressor order of this time series is 2 (it is generated from 2 past values). Let us note the lack of a x_{t-1} term, as well as the presence of a noise ε_t (about 10% of the maximum value of the series). The series does not contain any exogenous variable ($P = 0$), and is shown in Figure 2.

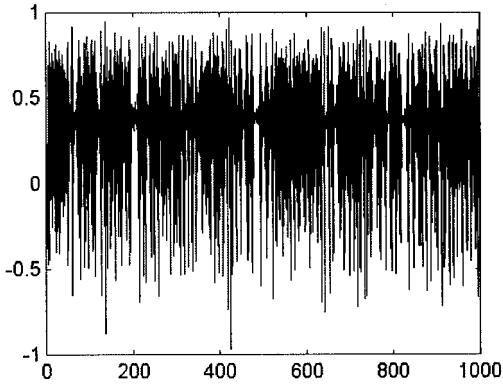

Fig. 2. *Artificial time series generated according to equation (4).*

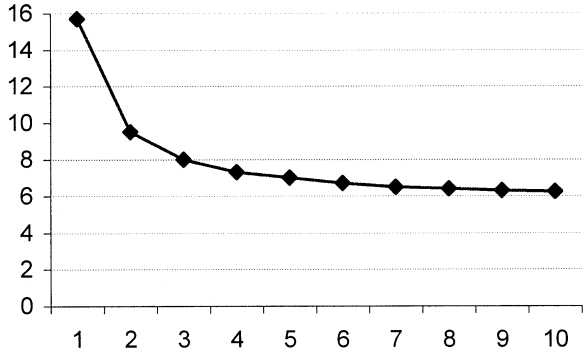

Fig. 3. *Sum of quadratic errors (on 1000 test points) obtained with an AR model for different values of the regressor order.*

We begin by looking at the results of a forecasting by a linear (auto-regressive - AR) model on this series. Figure 3 shows the sum (on 1000 points) of the quadratic errors obtained with a linear AR model of increasing order. Obviously, the error decreases with the order. However, it is also evident from Figure 3 that a linear model cannot catch the non-linear dynamics with a low regressive order (although only two past values are used to build the series).

We then proceed by using the non-linear methodology described above. To ensure that the whole dynamics of the series is collected, we build an initial regressor matrix of order 6. The Grassberger-Procaccia algorithm to estimate the intrinsic dimension of the series gives 2.12, which is close to the exact value 2. Note that the noise ε_t added to the series inevitably increases the intrinsic dimension.

The following step of the method is the projection of the set of the points (rows of the regressor matrix) from R^6 to R^2. The dimension of the final regressor vector is thus 2.

In a next step we uses this 2-dimensional regressor as input to a non-linear prediction model. As an example, we use a Multi-Layer Perceptron with one hidden layer and five hidden units; other non-linear models could be used. The sum of quadratic errors obtained with this MLP is around 5 (on 1000 points), which is significantly lower than the errors illustrated in Figure 3 (linear model).

We also compare this result to the error obtained with a similar Multi-Layer Perceptron, where the input vector is the set of N last values from the raw series. Figure 4 shows this error as a function of N. The horizontal dotted line corresponds to the error obtained with our method; we conclude that we obtain (for this example) an error similar to a result obtained by trial and error on several non-linear (MLP) models, which was the goal of our investigation. This easiness of implementation will be valuable when dealing with a “real-size” data set for which the non-linear regressor order is unknown.

4. Application to the BEL 20 stock market index

An interesting example of time series in the field of finance is the Belgian Bel 20 index. As stressed in the introduction, the application of time series forecasting to financial

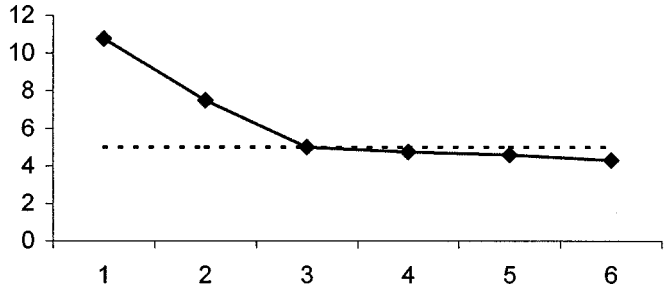

Fig. 4. *Sum of quadratic errors (on 1000 points) obtained with a MLP network for different values of the regressor order. The horizontal line corresponds to the result of the proposed method.*

market data is a real challenge. The efficient market hypothesis (EMH) remains up to now the most generally admitted one in the academic community, while essentially challenged by the practitioners. Under EMH, one of the classical econometric tools used to model the behaviour of stock market prices is the geometric Brownian motion. If it does represent the true generating process of stock returns, the best prediction that we can obtain of the future value is the actual one (they follow a random walk). Results presented in this section must therefore be analysed with a lot of caution.

To succeed in determining the variations of the Bel 20 index, other variables that could be of influence are included as inputs (exogenous variables). We selected international indices of security prices (SBF 250, S&P500, Topix, FTSE100, etc), exchange rates (Dollar/Mark, Dollar/Yen, etc.), and interest rates (T-Bills 3 months, US Treasury Constant Maturity 10 years, etc.).

We used 2600 daily data of the Bel 20 index over 10 years to have a significant data set. The problem considered here is to forecast the sign of the variation of the Bel 20 index at time t+5, from available data at time t.

According to Refenes *et al.* (1997) and Burgess (1995), we use 42 technical indicators directly resulting from the inputs and the exogenous variables, for example:

- x_t , x_{t-10}, $x_{t-20,}$ x_{t-40}, …, y_t , y_{t-10}, …: returns;
- x_t - x_{t-5}, x_{t-5} - x_{t-10}, …, y_t - y_{t-5}, …: differences of returns;
- K(20), K(40), … : oscillators;
- MM(10), MM(50), … : moving averages;
- MME(10), MME(50), … : exponential moving averages;
- etc.

If we carry out a Principal Component Analysis (PCA) on these 42 variables, we note that 95% of the original variance is kept with the first 25 principal components: 17 variables can be removed without significant loss of information. The PCA is used to facilitate the subsequent processing by the CCA algorithm (lower computational load and better convergence properties).

The time series of the target variable, x_{t+5}, whose sign has to be predicted, is illustrated in Figure 5.

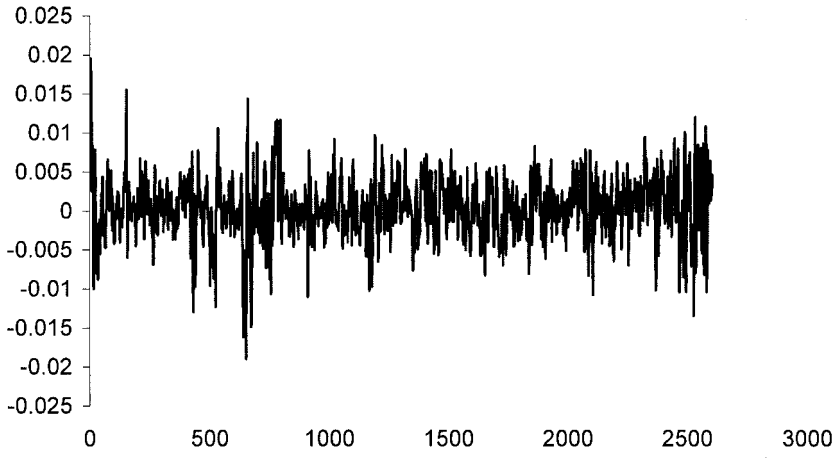

Fig. 5. *Time series of the target variable.*

This variable has to be predicted using the resulting 25 variables selected after PCA. The interpolator we use is a Radial-Basis Function (RBF) network as described in Section 2.1. Our interest goes to the sign of the prediction only, which will be compared to the real sign of the target variable.

The procedure described in Section 2.3 is used to estimate the intrinsic dimension of the data set; we obtain an approximate value of 9. We then use the CCA algorithm to project the 25-dimensional data (after PCA) on a 9-dimensional space. The RBF interpolator is used on the resulting 9-dimensional input vectors.

The network is trained with a moving window of 500 data. Each of these data consists in a 9-dimensional input vector (see above) and a scalar target (variation of the Bel 20 index). We use 500 data as a compromise between

- a small stationary set but insufficient for a successful training, and
- a large but less stationary training set.

For each window, the 500 input-target pairs form the training set, while the test set consists in the input-target pair right after the training set. This procedure is repeated for 2100 moving windows. On average, we obtain 60.3% correct approximations of the sign of the series on the training sets, and 57.2% on the test sets.

These results are encouraging. Moreover, it can be seen that better results are obtained during some periods and worst results during others. The first ones correspond to time periods where the series is more stationary. Figure 6 represents a moving average on 90 days on the results of the prediction. It clearly shows that the prediction results themselves do not form a random series: when the forecasting is correct over several consecutive days, the probability that it will be correct at the next time step is high.

To quantify this idea, we filter the results with the following rule. We look at the average of sign predictions (correct – not correct) over the last 5 days. If this average increases or remains constant at time t, then we keep the forecasting at time t+1. If it decreases, then we disregard the forecasting at time t+1. With this method, we keep 75.4% of the forecasts; the average score of correct prediction raises to 65.3% (about

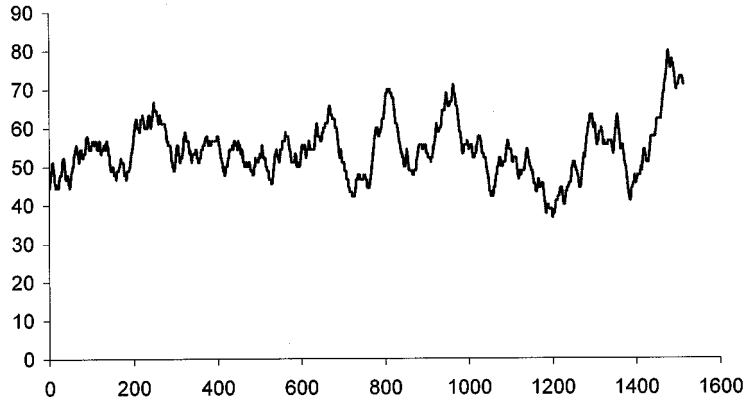

Fig. 6. *Percentage of correct approximations on a 90-days moving set.*

70% of increases and 60% of decreases). This way of working a first attempt to use our mathematical procedure in a real-world financial context.

5. Conclusion

We developed a method to predict time series with non-linear tools. The specificity of the method is:

- to use as much information as possible as input to the model (many past values of the series, many exogenous variables);
- to compress this information (by a non-linear method) in order to obtain a state vector of limited size, facilitating the subsequent regression and the generalization ability of the forecasting algorithm;
- to fit a non-linear regressor (here a RBF neural network) on the reduced vectors.

We show that this method is able to find non-linear relationships in artificial and real-world financial series. On a difficult task, which consists in forecasting the tendency of the Bel 20 stock market index, we show that this method improves the results compared both to linear models and to non-linear ones where the non-linear compression is not used.

From the financial point of view, the results seem sufficiently strong to question the random walk hypothesis (such conclusions are, for example, also reached by Brock *et al.* (1992) by using a set of classical technical analysis rules). However a lot of work remains to be done. In particular, the statistical significance of the results should be carefully evaluated; a bootstrap procedure like the one proposed by Sullivan *et al.* (1999) (reality check bootstrap) to take into account the real danger of data snooping (Lo, 1990) could be used. From a theoretical point of view, it is not so clear that the results challenge the EMH hypothesis. Neural networks and independent component analysis are really new statistical tools that were not available to the financial communities (at least the Belgian one) during the analysed period. If the EMH hypothesis holds, the forecasting power observed during the past period (if proved to be statistically significant) would be washed out with its diffusion in the financial community.

References

Alligood K.T., Sauer T.D., Yorke J.A. (1997) *Chaos: An Introduction to Dynamical Systems*. Springer-Verlag, New York, pp. 537-556.

Bachelier L. (1914) *Le jeu, la chance, et le hasard*. Flammarion, Paris.

Box G.E.P., Jenkins G. (1976) Time series analysis: Forecasting and Control. Cambridge University Press.

Brock W., Lakonishok J., LeBaron B. (1992) Simple Technical Trading Rules and the Stochastic Properties of Stock Returns, *J. of Finance* XLVII, n°5, pp. 1731-1764.

Broomhead D.S., Lowe D. (1988) Multivariable functional interpolation and adaptive networks, *Complex Systems* **2**, pp. 321-355.

Burgess A.N. (1995) Non-linear Model Identification and Statistical Significance Tests and their Application to Financial Modelling. In: Artificial Neural Networks, *Inst. Elect. Eng. Conf.*

Campbell J., Lo A., MacKinley A. (1997) *The Econometrics of Financial Markets*, Princeton University Press.

Demartines P., Hérault J. (1997) Curvilinear Component Analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Trans. on Neural Networks* **8** (1), pp. 148-154.

Edwards R., Magee J. (1988) *Technical Analysis of Stock Trends*, 5th edn., John Magee, Boston.

Fama E.(1965) The Behavior of Stock Market Prices, *J. Business* **38**, pp. 34-105.

Grassberger P., Procaccia I. (1983) Measuring the Strangeness of Strange Attractors, *Physica D* **56**, pp.189-208.

Kohonen T. (1995) Self-organising Maps, *Springer Series in Information Sciences*, 30, Springer, Berlin.

Ljung L. (1987) *System Identification: Theory for the user*. Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Lo A. (1990) Data-Snooping Biases in Tests of Financial Asset Pricing Models, *Review of Financial Studies* **3**, pp. 431-468.

Refenes A.N., Burgess A.N., Bentz Y. (1997) Neural Networks in Financial Engineering: A Study in Methodology. *IEEE Transactions on Neural Networks* **8** (6), pp. 1222-1267.

Rumelhart D., Hinton G., Williams R. (1986) Learning representation by back-propagating errors, *Nature* **323**, pp. 533-536.

Sjoberg J., Zhang Q., Ljung L. *et al.* (1995) Nonlinear black box modeling in system identification: A unified overview, *Automatica* **33**, (6), pp.1691-1724.

Sullivan R., Timmermann A., White A. (1999) Data-Snooping, Technical Trading Rule Performance, and the Bootstrap, *J. Finance* (to be published).

Takens F. (1985) On the numerical Determination of the dimension of an attractor. In: *Lecture Notes in Mathematics*, 1125, Springer-Verlag, pp. 99-106.

Theiler J. (1990) Statistical Precision of Dimension Estimators, *Phys. Rev. A* **41**, pp. 3038-3051.

Verleysen M., Hlavácková K. (1994) An Optimized RBF Network for Approximation of Functions. In: *Proc of European Symposium on Artificial Neural Networks*, Brussels (Belgium), D facto publications (Brussels).

Weigend A.S., Gershenfeld N.A (Eds) (1994) *Time series Prediction: Forecasting the Future and Understanding the Past*, Reading, MA: Addison Wesley.

Werbos P. (1974) Beyond regression: new tools for prediction and analysis in the behavioral sciences PhD thesis, Harvard University.