Decision Support Systems (Elsevier Science)
Accepted for publication, March 2003

EDDIE-Automation, a Decision Support Tool for Financial Forecasting

Edward Tsang[1], Paul Yung[2] & Jin Li[3]
[1] Department of Computer Science, University of Essex, Colchester CO4 3SQ UK
[2, 3] Freelance programmer
Ex-student, Department of Computer Science, University of Essex, Colchester UK
Contact: edward@essex.ac.uk; http://cswww.essex.ac.uk/CSP/finance

Abstracts

EDDIE is a genetic programming based decision support tool for financial forecasting. EDDIE itself does not replace forecasting experts. It serves to improve the productivity of experts in searching the space of decision trees, with the aim to improve the odds in its user's favour. The efficacy of EDDIE has been reported in the literature. However, discovering patterns in historical data is only the first step towards building a practical financial forecasting tool. Data preparation, rules organization and application are all important issues. This paper describes an architecture that embeds EDDIE for learning from and monitoring the stock market.

Key words: financial forecasting tools, genetic programming

Biographical notes:

Dr Edward Tsang is a professor in computer science at the University of Essex. He specializes in applied artificial intelligence, especially heuristic search, stochastic methods and their application to constraint satisfaction.

Paul Yung received his Master degree from the Department of Electronic Systems Engineering, University of Essex. He is a freelance programmer specialized in telecommunication and Unix systems.

Dr Jin Li received his PhD in Computer Science from University of Essex. He is a freelance programmer specialized in Window and web-based software design.

1. Background

This paper describes a decision support tool for financial forecasting. Suppose, through diligent research, one discovers that a particular ratio, alongside with other commonly known factors (such as interest rate, money supply, dividends, moving average and volatility), is highly relevant to the movement of a share, but do not know the exact mathematical relation. The question is: could this discovery give this person any advantage in trading? In other words, how could one take advantage of such knowledge? The answer is: discovering relevant factors alone does not offer too much help. What needs to be done further is non-trivial. Even if one limits oneself to using decision trees as in ID3/C4.5/See5/C5.0 [17] [18] or genetic programming [7] [10] [11], one still needs to search in a huge space of possible decision trees. The search space is exponential in size to the maximum depth that one may want to look at and it is quite impossible to search exhaustively for useful trees, even with the help of fast computers at today's standard [11].

EDDIE (which stands for *Evolutionary Dynamic Data Investment Evaluator*) is an interactive tool, based on genetic programming (GP), to help analysts to search the space of interactions between factors (in the form of decision trees) and make financial decisions. As the aim of this paper is to describe the role of EDDIE in a decision support system (as opposed to describing its forecasting ability, which has been reported in [12] [14] [24], and will be briefly summarized below), a survey of scientific financial forecasting methods will not be presented (see [3] [7] [8] [14] [19] for a few examples)

here.

The contribution of EDDIE is in effectively searching for combinations (interactions) of financial indicators and discovering thresholds (to be elaborated in the next section). However, after developing a system that can mine patterns in the data, many issues are left to be studied in building a practical financial forecasting tool. For example, one must study *how* such predictions, which are at best statistically reliable (i.e. there is no guarantee that they will be correct every time), for investment. This is *not* the scope of this paper. In this paper, we investigate the practical issues in generating and using such predictions. In the remaining part of the paper, we shall describe an architecture that employs EDDIE in learning from and monitoring the stock market. We shall argue that the implemented automation system enables the user to conduct large scale learning and monitoring in the stock market efficiently.

2. EDDIE for financial forecasting

EDDIE is a decision support tool based on genetic programming [9] [10] [11]. The efficacy of EDDIE has been reported in [5] [13] [14] [21] [23] [24]. A brief summary is given in this section.

The first implementation of EDDIE, called EDDIE-1, was applied to horse racing [21]. It's performance was promising but limited, due to the lack of data in horse racing at the time. FGP-1 (co-named EDDIE-3; FGP stands for *Financial GP*), an implementation of EDDIE, was applied to financial forecasting [12]. One of the main applications of EDDIE was to generate and test hypotheses of the following form:

"Will the price of share/index X rise by r% within the next n trading days?"[1]

For example, one may want to investigate whether the price of the FTSE-100 index will rise by 2.2% within the next month. The syntax of the trees to be generated is as follows:

```
            Tree  :=  "If-then-else"  <Condition>  <Tree>  <Tree> | <Prediction>
     <Condition>  :=  <Condition> "And" <Condition>  |  <Condition> "Or" <Condition>  |
                      "Not" <Condition>  |  <Indicator> <RelationOperation> <Threshold>
     <Indicator>  :=  "MV_12"  |  "MV_50  "  |  "Filter_5"  |  "Filter_63"|  "TRB_5  "  |  "TRB_50"  |
                      "Vol_12" |  "Vol_50
<RelationOperation>  :=  ">"   |   "<"   |  "="
     <Threshold>  :=  Number
     <Prediction>  :=  "Positive" | "Negative"
```

Figure 1 – The BNF grammar used by FGP

A "Positive" prediction means that the goal can be achieved; "Negative" means otherwise. An example decision tree (annotated for readability) is shown below:

```
(IF  (MV_50  <  -18.45) THEN  Positive
   ELSE (IF  ((TRB_5  >  -19.48) AND (Filter_63 < 36.24)) THEN Negative ELSE Positive))
```

This decision tree actually represents three simple rules: (a) if MV_50 is less than –18.45, then positive is predicted; (b) if MV_50 is not less than –18.45, and TRB_5 is greater than –19.48, and Filter_63 is less than 36.24, then negative is predicted; and (c) otherwise (not elaborated here) positive is predicted.

In GP terms, FGP-1 uses {If-then-else, And, Or, Not, <, =, >} as functions and

[1] In fact, one can ask whether prices will *fall* by r% within the next n days, or, more generally, within the next n units of time where each row in the data set represents information at each unit of time. In [15],

indicators, numbers and predictions as terminals. The crossover operator was designed to take care of the type of the branches. Indicators were taken from the finance literature, such as [1] [4] [6] [18], normalized. For example, MV_12 is the 12-days moving average divided by the mean in the last 12 days. Vol_50 is the standard deviation of the prices in the last 50 days divided by the last 50 days' mean. Details of the parameters, which are not central to the ideas in this paper, can be found in [13].

EDDIE is useful to its users because even if the indicators are relevant to the prediction of the prices, finding (a) the logical interaction between the various indicators and (b) the thresholds (e.g. -18.45 in the above example) could be extremely laborious without tools such as genetic programming and neural networks.[2] There is no magic behind genetic programming or neural networks. All they do is to efficiently explore the space of possible patterns.

FGP-2 (co-named EDDIE-4) extended FGP-1 by using a constrained fitness function, which allows the user to adjust the level of cautiousness [24]. When the user instructs FGP-1 to recommend cautious rules, FGP-2 will attempt to improve precision, possibly at the price of missing opportunities. This means FGP-2 will make fewer recommendations to buy (or sell). The effectiveness of FGP-2 as a forecasting tool has been reported in [13] [14] [24]. FGP-2 is designed as a forecasting tool, in which the user is given control over the percentage of opportunities to pick up [24] (as opposed to incorporating a *misclassification cost* [18] in the objective function).

EDDIE was used to monitor arbitrage opportunities by finding patterns in tick data.

[2] Neural networks are perfectly capable of finding nonlinear patterns [7], though they are considered by some as black-boxes.

3. EDDIE-Learn 1.0, an automated learning architecture

Finding patterns in a given set of data is not the only laborious task in forecasting. The automation of the learning process is a necessary step towards turning EDDIE into an effective forecasting support tool. Here are some of the reasons:

1. Data preparation plays an important part in data mining. One does not always know which indicators are relevant to the movement of a particular share. One often has to prepare different sets of indicators and data-mine in them.
2. One often needs to experiment with a wide variety of hypotheses in order to feel one's way in the data.
3. To apply GP effectively, one has to experiment with different parameter sets, such as the population size, number of iterations and level of cautiousness.

For example, over a dozen versions of data sets and various hypotheses have been experimented and a number of hypotheses have been tested when we applied EDDIE to finding arbitrage opportunities [14]. Preparation of data took hours of laborious manual and programming work.

To serve the above needs, an EDDIE-based automatic learning system has been implemented. Input to the system includes:

(a) Data files, in comma-separated values (CSV) format; in the implemented system, this includes the date and share prices. It is possible to include new attributes in the data;

(b) The range file, in text format, which defines the range of data to be used for training and testing;

(c) The control file, in text format, which defines the GP parameters, including population size, mutation rate, etc.

(d) Program parameters: the rate of return (r), forecasting horizon (n) and the precision threshold (p). Together, they define the target return (r% within the next n days) and the minimum precision (on the test data) for the rules to be accepted.

For each run, a report in CSV format (which is suitable for spreadsheet use as well as electronic processing) is generated. The main output of EDDIE-Automation 1.0 is a collection of rules with precision above the threshold specified (default is 75%). Intermediate data files are generated for verification purpose.

Figure 2 gives an overview of the EDDIE-based automatic learning system, EDDIE-Learn 1.0. To use the system, the user stores the data files of all the shares that he/she wants EDDIE to learn from in the directory dir_input. Each data file contains a time series, in the form of time (e.g. dates) and price pairs. The program autofeedfgp completes the rest of the training by invoking a sequence of programs: The program autofeed calculates the indicators for each file in dir_input – after adding extra columns to the table, the augmented data will be stored in the directory dir_cal. The program autochop selects the rows specified in the parameter file range.txt (explained in point (b) above). The data prepared is fed into an automation of EDDIE, FGP-3. FGP-3 also reads in two parameter files, range.txt and control.txt (explained in point (c) above).

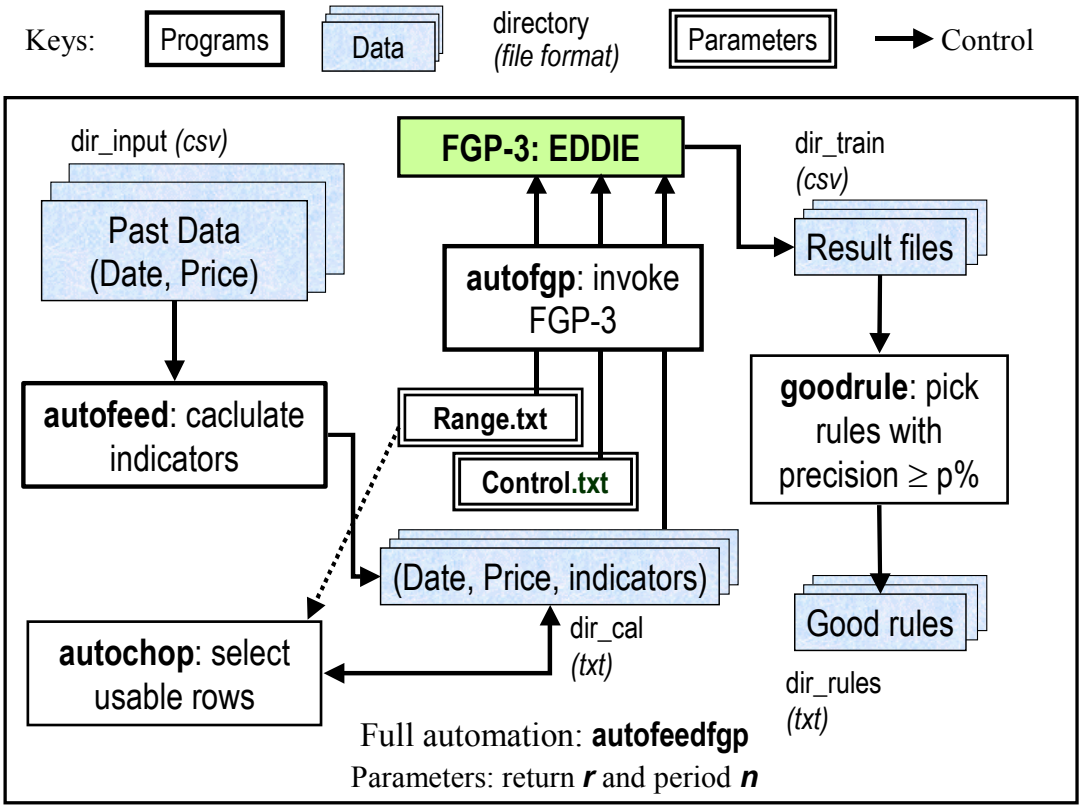

Figure 2 – EDDIE-Learn 1.0, the Learning Architecture that embeds EDDIE

Training results are stored in a directory (called dir_train) in CSV format (to enable users to inspect them with a spreadsheet). The program goodrule picks the rules which performance is above the specified precision threshold *p* (explained in point (d) above), and put them in a good-rules bank (dir_rules). Care has been taken in generating the names of the files, which reflect the return (*r*), period (*n*) and precision (*p*). Multiple rules may be generated for the same share – this leaves us with the option to combine rules and recommendations in the future.

4. EDDIE-Automation 1.0, an automated monitoring architecture

In the previous section, we have explained how EDDIE-Learn 1.0 learns patterns and put them into a good-rules bank. In this section, we present EDDIE-Automation 1.0, an architecture that automates the whole learning and monitoring process. The overview of the architecture is shown in Figure 3.

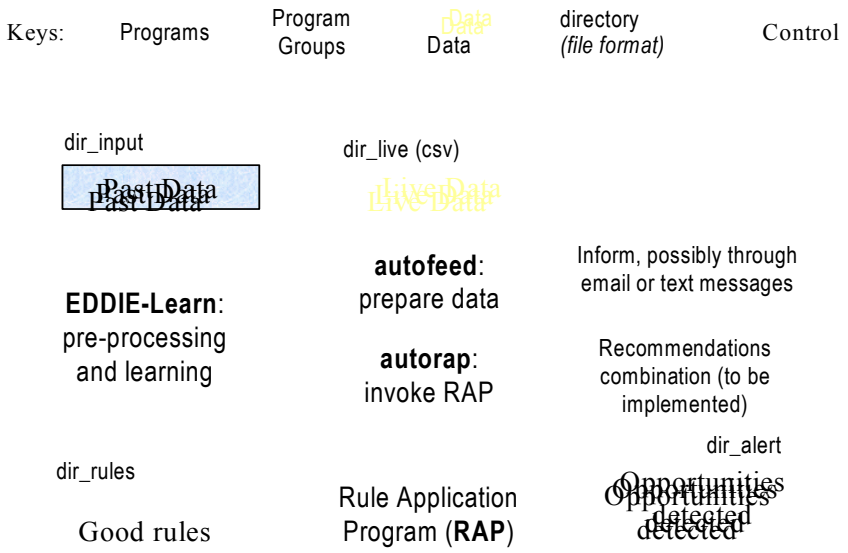

Figure 3 – The EDDIE-Automation Architecture

In order to apply the rules learned, a Rules Application Program (RAP) is implemented. Given (a) a decision tree DT and (b) a record R, which comprises the price and the indicators that DT used in its training, RAP returns "True" if R indicates a positive position according to DT (meaning that *r*% can be achieved within the next *n* days for the *r* and *n* specified in the parameter file range.txt), and "False" otherwise.

The decision trees in the good-rules bank are used to monitor their corresponding

shares in order to detect investment opportunities. To facilitate that, live data is fed into the directory dir_live in Figure 3. The program autofeed (which is also used in EDDIE-Learn 1.0, see Figure 2) prepares the data for RAP. This will provide RAP with a record with the prices and other indicators necessary for its calculation. The program autorap picks up each rule in the good-rules bank in turn, and apply it to the corresponding record prepared by autofeed (correspondence is established through their file names). The results of all predicted investment opportunities are stored in a directory called dir_alert. All investment opportunities in dir_alert are collected and summarized in a text file, which can be delivered to investors by email or text messages by phone if desired.

In the good-rules bank, multiple rules may apply to the same share. Recommendations may or may not agree with each other. In the future, we would consider combining recommendations before communicating with the users. There are many ways to do so. For example, one may only decide to buy a particular share if at least *m* out of the *n* rules in the good-rules bank make positive recommendations. Or, one may like to generate rules for both long-term and short-term trading, and only act if both long-term and short-term prospects are positive. Different ways of combining rules are on the agenda of the EDDIE project.

EDDIE-Automation 1.0 was built in modules. This allows flexibility in the system. For example, should the input files take a different format, or should the user require a different set of indicators, only the autofeed module needs to be changed. Even the core module FGP-3 could be replaced by another learning system that uses the same input and output format (such as C4.5 [17] and See5/C5.0 [18]).

All parts of EDDIE-Automation 1.0 shown in Figure 3 have been implemented except live data-feed, combining recommendations and communication with the user. In the system implemented, live data is placed into a directory (dir_live in Figure 3) manually. FGP-3 was implemented in C++. RAP was implemented in Java. The rest of the programs were written in Perl script. All experiments were run under Windows 2000.

5. EDDIE-Automation 1.0 as a Decision Support System

First, it is worth noting that although the implementation generates technical rules only, the EDDIE-Learn architecture is not limited to technical analysis (although there are good arguments for studying technical analysis scientifically [6] [16]). One could also use indicators that are generated by economic models. This was done in EDDIE-ARB, where the theoretical price of options was also used for detecting arbitrage opportunities [14].

Secondly, it is worth re-iterating that EDDIE-Automation is no replacement for experts. EDDIE does not guarantee 100% correctness in its predictions. Instead, it aims to improve the odds in its user's favour. It helps to find patterns in past data, which can be used for reference by its user. It cannot predict new events or market crashes, unless its user can. (In general, EDDIE has no prediction power unless the user does.) What EDDIE can do is to significantly enhance the user's productivity in finding patterns and monitoring the market. As a decision support tool, EDDIE-Automation should be judged by how and how much it could improve the user's productivity. We shall therefore look at various aspects of financial data mining below.

Data preparation is crucial to data mining. It is often labour-intensive too. One often needs to experiment with different ways to create new attributes (such as new ratios), remove attributes (in order to reduce the search space), remove records (to reduce noise or reduce complexity), etc. before feeding data into any learning system. Data preparation often involves domain knowledge. While EDDIE-Learn 1.0 cannot offer much help in this aspect, it provides a framework to automate data preparation once the user has decided what indicators to create, which columns or rows to use for training and testing, etc. By supplying EDDIE-Learn 1.0 with an appropriate module to prepare tables from raw data, users can experiment with a large number of data sets automatically.

Where EDDIE-Learn 1.0 can provide significant help to its user is in hypotheses testing. Users may ask questions of the format "will any of these shares rise by *r*% within the next *n* days?". After the user stores the price series of all the shares of interest into the input directory (dir_input in Figure 2), a simple instruction will set EDDIE-Learn 1.0 off to learn rules for all the shares in that directory. Users may easily repeat the experiments by varying *r* and *n*.

Going back to the points made in Section 1, EDDIE helps the users to discover interactions between financial indicators in the form of decision tree. Each branch in the decision tree represents the combination of a number of indicators. It may be the case that indicator X is only relevant if indicator Y is also considered. EDDIE will also search for threshold values. Given the exponential size of search space, it is in practice impossible for human users to discover such thresholds manually. There is no guarantee that EDDIE will find patterns, but it stands a better chance to do so than its human user as it can work

day and night continuously.

When FGP-3 is asked to act cautiously, each decision tree in the good-rules bank may pick up only a small number of investment opportunities. This is not a serious problem, as one can use EDDIE-Learn 1.0 to learn as many rules as one's computation power permits – with the hope (but no guarantee) that different opportunities will be picked up by different rules. With a large number of rules in the good-rules bank, it is necessary to automate the monitoring process. EDDIE-Automation 1.0 provides a framework to monitor the market and report to the users when opportunities are detected.

Obviously, the more shares and indices that one studies, the more investment opportunities one could potentially discover. Besides, the more rule sets one generates, the more chance that one could spot investment opportunities or combine recommendations. EDDIE-Automation 1.0 enables the users to conduct large-scale learning to fill the good rules bank (directory dir_rules in Figure 2 and Figure 3). EDDIE-Automation 1.0 is a useful tool because the amount of work involved, in both learning and monitoring, is huge and therefore beyond human effort.

To summarize, this paper describes the role of EDDIE in a decision support system. EDDIE enables the user to discover patterns that he/she could not practically discover manually. EDDIE-Learn 1.0 enables the user to do large-scale learning. EDDIE-Automation 1.0 is a practical tool for forecasting. Finally, it is worth re-iterating that EDDIE is only a useful decision support tool, not a replacement for forecasting experts.

Acknowledgements

The EDDIE project was started by James Butler, who implemented EDDIE-1 and tested it on horse racing. Many colleagues and students at University of Essex have contributed to the EDDIE project through discussion. The project is grateful to the support by Dr Sheri Markose, Director of Institute for Studies in Finance at University of Essex. This project is partly supported by two Research Promotion Funds by the University. Jin Li was supported by the Overseas Research Scholarship and the University of Essex Scholarship.

References:

[1] Alexander, S.S., *Price movement in speculative markets: Trend or random walks, No. 2*, in Cootner, P. (ed.), the random character of stock market prices, MIT Press, Cambridge, MA, 1964, 338-372

[2] Angeline, P. & Kinnear, K.E.Jr. (ed.), *Advances in genetic programming II*, MIT Press, 1996

[3] Bauer, R.J.Jr., *Genetic algorithms and investment strategies*, John Wiley & Sons, Inc., 1994

[4] Brock, W., Lakonishok, J. & LeBaron, B., *Simple technical trading rules and the stochastic properties of stock returns*, Journal of Finance, 47, 1992, 1731-1764

[5] Butler, J.M., *EDDIE beats the market, data mining and decision support through genetic programming*, Developments, Reuters Limited, Vol.1, July 1997

[6] Fama, E.F. & Blume, M.E., *Filter rules and stock-market trading*, Journal of Business 39(1), 1966, 226-241

[7] Goonatilake, S. & Treleaven, P. (ed.), *Intelligent systems for finance and business*, Wiley, New York, 1995

[8] Kasabov, N. K., *Foundations of neural networks, fuzzy systems and knowledge engineering*, MIT Press, 1996

[9] Koza, J.R. *Genetic Programming: on the programming of computers by means of natural selection*. MIT Press, 1992

[10] Koza, J., Goldberg, D., Fogel, D. & Riolo, R. (ed.), *Procedings, First Annual Conference on Genetic programming*, MIT Press, 1996

[11] Langdon, W.B. & Poli, R., *Foundations of genetic programming*, Springer, 2001

[12] Li, J. & Tsang, E.P.K, *Investment decision making using FGP: a case study*, Proceedings of Congress on Evolutionary Computation (CEC'99), Washington DC, USA, July 6-9 1999.

[13] Li, J., *FGP: a genetic programming based tool for financial forecasting*, PhD Thesis, University of Essex, UK, 2001

[14] Mahfoud, S. & Mani, G., *Financial forecasting using genetic algorithms*, Applied Artificial Intelligence, Vol.10, 1996, 543-565

[15] Markose, S., Tsang, E. & Er, H., *EDDIE for stock index options and futures arbitrage*, in Chen, S-H. (ed), Genetic algorithms and genetic programming in computational finance, Kluwer Academic Press, 2002, 281-308

[16] Neely, C., Weller, P. & Ditmar, R., *Is technical analysis in the foreign exchange market profitable? a genetic programming approach*, in Dunis, C. & Rustem, B. (ed.), Proceedings, Forecasting Financial Markets: Advances for Exchange Rates, Interest Rates and Asset Management, London, May 1997

[17] Quinlan, J.R., *C4.5: programs for machine learning*, Morgan Kaufmann, San Mateo, 1993

[18] Quinlan, J.R, *Data mining tools See5 and C5.0*, http://www.rulequest.com/see5-info.html (accessed 3 February 2003)

[19] Sornette, D. and Zhou, W.-X., *The US 2000-2002 Market Descent: How Much Longer and Deeper?* Quantitative Finance 2 (6), 2002, 468-481

[20] Sweeney, R.J., *Some new filter rule test: Methods and results*, Journal of Financial and Quantitative Analysis, 23, 1988, 285-300

[21] Tsang, E.P.K., Li, J. & Butler, J.M., *EDDIE beats the bookies*, International Journal of Software, Practice & Experience, Wiley, Vol.28 (10), 1998, 1033-1043

[22] Tsang, E.P.K. & Li, J., *Combining Ordinal Financial Predictions With Genetic Programming*, Proceedings, Second International Conference on Intelligent Data Engineering and Automated Learning, Hong Kong, December 13-15, 2000

[23] Tsang, E.P.K., Li, J., Markose, S., Er, H., Salhi, A. & Iori, G., *EDDIE In Financial Decision Making*, Journal of Management and Economics, http://www.econ.uba.ar/www/servicios/publicaciones/journal4/Index.htm, Vol.4, No.4, November 2000

[24] Tsang, E.P.K. & Li, J., *EDDIE for financial forecasting*, in Chen, S-H. (ed.), Genetic Algorithms and Programming in Computational Finance, Kluwer Academic Publishers, 2002, 161-174