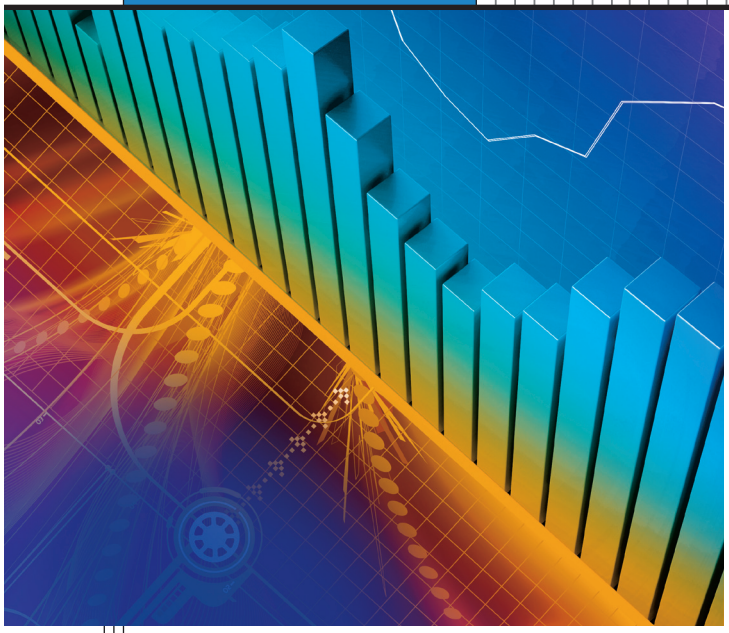

COMPUTATIONAL FINANCE

Chaiyakorn Yingsaeree and Philip Treleaven, *UK Centre for Financial Computing, London*
Giuseppe Nuti, *Citadel Securities, New York*

Banks and investment funds are increasingly basing their competitiveness on the quality of their quantitative technology, including programming techniques, analytical methods, and applications such as financial forecasting, option pricing, and risk management, which are all essential elements of the field of computational finance.

Banks and investment funds increasingly base their competitiveness on the quality of their quantitative technology—programming, analytics, and financial applications—a trend that has made computational finance crucial.

Computational finance is a cross-disciplinary field that focuses on the financial services industry and relies on mathematical finance, numerical methods, and computer simulations to make trading, hedging, and investment decisions, and also facilitates portfolio risk management. Significant overlap exists with related areas, such as financial modeling, mathematical finance, and financial engineering.

- *Financial modeling,* the most general of the related terms, covers financial computations such as option pricing, with the central aim of modeling valuation under uncertainty.
- *Mathematical finance* is the branch of applied mathematics concerned with financial markets. Traditionally associated with stochastic calculus, in practice this discipline spans various areas of applied mathematics.
- *Financial engineering* focuses on innovation, aiming to produce new securities, such as options and futures markets derivatives.

Table 1 shows a simple computational finance taxonomy.[1] A subdomain of computational science,[2] computational finance consists of two distinct branches. *Data mining* is a knowledge discovery technique that extracts hidden patterns from huge quantities of data, enabling formation of hypotheses. *Computer modeling* provides simulation-based analysis that predicts system dynamics to test the validity of an underlying assumption.

TOOLS OF THE TRADE

Table 2 lists the statistical and artificial intelligence techniques that are widely used in computational finance. Additional disciplines used in computational finance include symbolic and algebraic computing, numerical analysis, computational geometry, and visualization and graphics.

Symbolic and algebraic computing

Symbolic and algebraic computing (SAC) is concerned with representing and manipulating information in symbolic forms, such as mathematical expressions. SAC is typically used to perform analytical manipula-

 Published by the IEEE Computer Society 0018-9162/10/$26.00 © 2010 IEEE

Table 1. Computational finance taxonomy.

Analytical method	Programming techniques	Finance applications
Classification	*Rule-based methods:* decision-tree learning, first-order learning *Geometric methods:* neural networks, support vector machine *Probabilistic methods:* naïve Bayes classifiers, maximum entropy classifiers *Prototype-based methods:* nearest-neighbors classification	Stock selection Bankruptcy prediction Bond rating Fraud detection
Optimization	Simulated annealing, genetic algorithms *Dynamic optimization:* dynamic programming, reinforcement learning *Static optimization:* simplex methods, interior-point methods	Portfolio selection Risk management Asset liability management
Regression	*Dictionary representation:* linear regression, polynomial estimates, wavelet regression, neural networks *Kernel representation: k*-nearest neighbors, support vector machines	Financial forecasting Option pricing Stock prediction
Simulation	*Stochastic simulation:* Markov chain Monte Carlo simulations *Agent-based simulation:* genetic algorithms, genetic programming	Option pricing Market microstructure

tion in the early problem-solving steps of specification and model creation. This preprocessing usually leads to better understanding of the mathematical problem, important simplifications, and selection of proper solution procedures.

Numerical analysis

This discipline focuses on creating, analyzing, and implementing algorithms with continuous variables that originate from algebra, geometry, and calculus. For example, analyses can focus on solving both linear and nonlinear equations, differential and integral equations, as well as approximating a given function's integrals and derivatives. Numerical analysis techniques usually provide high-precision approximations rather than exact solutions.

Computational geometry

A branch of computer science, computational geometry focuses on the design, analysis, and implementation of algorithms for solving geometric problems. Examples include range searching (preprocessing a set of points to efficiently identify that they occupy a specified query region), nearest neighbor (preprocess a set of points to efficiently find which point lies closest to a query point), and ray tracing (given a set of objects in space, produce a data structure that efficiently tells which object a query ray intersects first).

Visualization and graphics

These disciplines cover techniques for creating images, diagrams, or animations to communicate both abstract and concrete ideas. For example, they let users perceive patterns and relationships that might be missed in tables of numbers. Traditionally, researchers applied visualization techniques primarily while pre- or postprocessing solutions. A recent advance—using visualization tools to observe and steer the computation during runtime—helps depict sparse matrices in the context of matrix laboratories or other numerical libraries.

Computational statistics

Computational statistics[3] addresses methodologies in which intensive computing is an integral component, usually at the interplay between computer science and data analysis. Built on the mathematical theory and methods of statistics, the discipline includes visualization, statistical computing, and Monte Carlo methods. The emphasis in computational statistics is often on exploratory methods. These include statistical inference methods such as resampling using bootstrap and jackknife techniques, Markov chain Monte Carlo algorithms, and regression and classification.

Probability density estimation is a subdivision of computational statistics that covers methods of constructing estimates for unobservable underlying probability density functions from observed data. Examples include kernel density estimation, nearest-neighbor estimation, and maximum-penalized-likelihood estimation.

Another subdivision, statistical inference, makes inferences concerning some unknown aspect of a population from a random sample. This can include both point and interval estimation. These inferences can be made using traditional most-frequent methods, such as the maximum likelihood method and the minimum-mean-squared error, or Bayesian methods such as Markov chain Monte Carlo algorithms and the Kalman filter.

Statisticians also use regression methods to model and analyze a relation between a dependent variable and one or more independent variables. Although it has a strong connection to supervised learning techniques in artificial intelligence, the main focus in statistics is seeking to determine the relation between the dependent and independent variables, not on obtaining the most accurate prediction of the dependent variable as supervised learning requires.

Table 2. Statistical and artificial intelligence techniques used in computational finance.

Statistical techniques	
Density estimation Constructing an estimate of an unobservable underlying probability density function from observed data	Kernel density Nearest-neighbor density Maximum penalized likelihood
Statistical inference Making inferences concerning some unknown aspect of a population from a random sample	**Point estimation** Frequentist approach: maximum likelihood, minimum-mean-squared error Bayesian approach: credible interval **Interval estimation** Frequentist approach: confidence interval Bayesian approach: credible interval
Regression Modeling and analyzing a relation between a dependent variable and one or more independent variables	Generalized linear model Generalize additive model
Artificial intelligence techniques	
Symbolic reasoning (knowledge-based) Symbolic inference and symbolic knowledge representation for use in making inferences	Rule-based expert systems Fuzzy logic Case-based reasoning Symbolic reasoning Constraint-based reasoning Multiagent systems Rule induction
Subsymbolic reasoning (machine learning) A machine's ability to improve its performance based on previous results	**Supervised learning** Constraint-based reasoning Support vector machines Bayesian learning Decision-tree learning Genetic classifiers Instance-based learning Multivariable regression Nearest-neighbor classifier Neural networks Genetic algorithms and genetic programming Artificial life Evolutionary systems Neuro-fuzzy systems **Unsupervised learning** Bayesian learning Support vector machines Genetic algorithms and genetic programming Neuro-fuzzy systems *K*-means clustering Self-organizing maps Principal components analysis **Reinforcement learning** *q*-learning Temporal difference learning Neurodynamic programming

The category includes the generalized linear and generalized additive models.

Artificial intelligence

AI techniques[4] subdivide into symbolic reasoning (logic, rule-based, and expert systems) and subsymbolic/machine learning (neural networks and evolutionary computation).

Symbolic (or knowledge-based) AI focuses on attempting to explicitly represent human knowledge in a declarative form that employs facts and rules. The classic model is a rule-based "expert" system comprising a

domain-specific knowledge base (the facts), a set of production rules, and an inference engine.

Subsymbolic (or machine learning) AI refers to a system that autonomously acquires and integrates knowledge. This capacity to learn from experience, analytical observation, and other means results in a system that can continuously self-improve, thereby offering increased efficiency and effectiveness.

Subsymbolic systems further subdivide into two categories. *Supervised learning* covers techniques used to learn the relationship between independent attributes and a designated dependent attribute, the label. Most induction algorithms fall into the supervised learning category. Examples include regression trees and discriminant function analysis. *Unsupervised learning* covers learning techniques that group instances without a prespecified dependent attribute. Examples include clustering algorithms, neural network (both supervised and unsupervised), self-organizing maps, and principal components analysis.

ANALYTICAL METHODS

The analytical methods used in computational finance comprise four broad categories: classification, optimization, regression, and simulation.

Classification

This analytical approach uses training examples, such as pairs of input and output targets, to find an appropriate function, with techniques broadly dividing into rule-based, geometric, probabilistic, and prototype-based methods.

Rule-based methods use a set of if-then rules to represent classifications. Geometric methods represent classification functions with a set of decision boundaries constructed by optimizing certain error criteria. Probabilistic methods use the Bayes theorem, which combines prior probabilities of classes and class-conditional densities from the instances to classify the object. Last, prototype-based methods use similarity between objects to decide on a good classification instead of directly constructing decision boundaries or relying on probability.

Classification uses computational statistics split into normal models—such as probit, logit, and generalized linear—and ordered and ordinal models (ordered probit and ordered logit), in which the category variable is an ordinal.

Symbolic AI approaches for classification include logic, rule-based systems (expert systems), and symbolic reasoning. Subsymbolic or machine learning AI approaches encompass support vector machines, k-nearest neighbors, Bayesian classifiers, and—more recently—evolutionary techniques such as the genetic algorithms used to find appropriate weights and topologies using neural network classifiers.

Many potential financial classification applications suggest themselves, such as classifying stocks into high and low returns, with the aim of selecting good stocks for investment. Likewise, classification techniques also can solve the task that classifies a bond as "counterparty credit rating," "default rating," or "issuer credit rating." Other applications include bankruptcy prediction and fraud detection.

Optimization

This category of analytical methods broadly subdivides into parametric and control optimizations. Analysts perform parametric optimizations to find the values for a set of variables that optimize the objective function. Conversely, control optimization finds a set of actions to be taken in different states to optimize some objective function of a system. Parametric optimization is often called static optimization because the solution is a set of static parameters for all states, while control optimization is called dynamic optimization because the solution depends on the state, which changes dynamically.

The capacity to learn from experience, analytical observation, and other means results in a system that can continuously self-improve, thereby offering increased efficiency and effectiveness.

In parametric optimization, if the problem considered has a linear objective function and linear constraints, statisticians can use numerical analysis techniques such as interior-point and simplex methods. However, when the objective function and constraints are nonlinear, statisticians frequently use stochastic search algorithms derived from AI, such as genetic algorithms and simulated annealing, to obtain an approximation of the optimal answer. Dynamic programming that gives optimal solutions traditionally provides control optimizations.

When the problem is complex, statisticians often turn to heuristic methods (inexact methods that produce solutions in a reasonable amount of computer time) to produce approximate solutions. Recently, reinforcement learning techniques (such as q-learning, temporal difference learning, and neurodynamic programming) from AI have evolved as efficient alternative methods to solve these problems when the number of states is manageable.

In finance, optimization methods have been used extensively to support investment decisions since 1952, when Howard Markowitz formalized the diversification principles in portfolio selection as an optimization problem. Optimizations also play an important role in financial risk management, which often takes the following form: optimize a performance measure (such as

expected investment return) subject to the usual operating constraints and the constraint that a particular risk measure for the companies' financial holdings does not exceed a prescribed amount.

Regression

Regression techniques classify the output function into dictionary and kernel representations. In dictionary representations, the output function is generally approximated by a linear combination of a set of basic functions. In kernel representations, the output function is a weighted sum of the data point in the training examples, where the weight is determined from a kernel function. These techniques can be further classified as adaptive and nonadaptive.

Regression methods use both computational statistics and artificial intelligence techniques. Statisticians have applied Bayesian methods, such as Bayesian linear regression and Gaussian process regression, to find the best function to describe the training data, using new information from training data to update prior beliefs about the underlying function. In artificial intelligence, a neural network is an example of an adaptive dictionary representation method. Other examples of these methods include support vector machines and *k*-nearest neighbor regression.

Stochastic simulation in computational statistics and agent-based simulation in AI are currently the two approaches most prominently used in finance.

A classic application of regression is financial forecasting, which uses related information to predict the future value of interesting entities—such as exchange rates, commodity prices, and stock prices. Analysts also can apply regression methods to model the relationship between financial instruments and their derivatives, especially in option pricing.

Simulation

Simulation models include stochastic and deterministic approaches based on the presence of random elements; static and dynamic models based on the significance of time; and discrete and continuous models. While time is not a significant variable in static models, dynamic models focus on the system's evolution over time. Dynamic models can be further classified as discrete or continuous. In discrete dynamic models, the system's state changes at discrete time intervals; in continuous models, the system's state changes continuously across time and is normally described by a set of differential equations.

Stochastic simulation in computational statistics and agent-based simulation in AI are currently the two approaches most prominently used in finance. Stochastic simulation uses a random number generator to generate random sample data based on some known distribution for numerical experiments. Agent-based simulation is an AI method in which multiple entities sense and stochastically respond to conditions in their local environments, which, as a whole, creates complex large-scale system behavior.

Over the past decade, simulation use has grown rapidly to encompass most subdisciplines of economic theory.[5] Researchers use stochastic simulation, particularly Markov chain Monte Carlo methods, to find numerical solutions for many stochastic processes in financial modeling, such as option pricing. They can use agent-based simulation to interact with heterogeneous agents and model complex phenomena in finance and economics—such as price oscillation and bubbles.

COMPUTATIONAL FINANCE APPLICATIONS

Three important examples of computational finance applications include financial forecasting, option pricing, and risk management.

Financial forecasting

In general, financial forecasting seeks to predict future values or behaviors for selected financial instruments, typically to support investment and trading decisions. Forecasting using computational methods combines elements of fundamental analysis—predictions based on economic factors and indicators, government policy, and societal factors—with technical analysis—predictions based on past market data, primarily price and volume.

Techniques. A forecasting problem can be formulated using either a classification or regression method.

Techniques for classification-based forecasting derive mainly from artificial intelligence: either expert systems, decision trees, neural networks, or evolutionary computation. Among these, evolutionary computation appears the most widely used, with many researchers reporting that evolutionary computation usually performs better than neural networks in classification-based forecasting.[6]

Computational statistics techniques are extensively used for regression-based forecasting. Traditionally, statistical approaches based on linear autoregressive analysis models were developed to analyze and forecast time series data. Although these statistical approaches provide reasonable accuracy over short periods, the accuracy of time series forecasting diminishes sharply as the prediction length increases or if the relationship between the variables is nonlinear. Consequently, nonlinear regression models, such as Gaussian processes and neural networks, typically yield better results.[7]

Case study. Predicting the movement of stocks provides a simple illustration of forecasting.

To achieve this, the first step is to specify the input entities (variables) and the desired output of the forecasting model. Inputs subdivide into fundamental and technical data. Fundamental data relates to the general economy (inflation and interest rates), the condition of the industry (the values of related indexes and the prices of related commodities), and the companies' actual activity (price-to-earnings ratio, debt ratio, and prognoses of future profits and sales). Technical data relates to the price and volume of stocks, such as the highest and lowest traded price during the day.

To forecast future values, the output should be continuous, and hence regression-based. Classification-based forecasting methods are more appropriate for predicting future behavior.

After specifying the model's input and output, the next step is to select the forecasting method and train the model using historical data.

To make this example concrete, consider the task of predicting the future movement of one particular stock.[8] The task here is to identify investment opportunities where a return of 2.2 percent or more can be achieved within the next 21 trading days. Thus, the model's output is positive when a return of 2.2 percent or more can be achieved and is otherwise negative.

The model's input is limited to technical indicators that consist of the moving average of the previous 12 and 50 trading days, the minimum price of the previous five and 63 trading days, and the maximum price of the previous five and 50 trading days.

Testing this model using historical prices from 7 April 1969 to 5 May 1980 generated 2,800 data cases.[8] Of these, we used 1,900 cases as training data, with the remaining 900 cases used as testing data. The output indicates that the resulting model can predict the opportunity 62.13 percent correctly.

Option pricing

An option is a financial instrument that gives its owner the right to trade another financial instrument for an agreed price at any time on or before a specified future date. The agreed price is termed the strike price, and the specified date is the expiration date. The asymmetric payoff that arises from the option holder's right to exercise the option in the future when it is profitable to do so makes the problem of placing a value on an option difficult.

Techniques. The two main approaches for valuing an option price are parametric or nonparametric methods. Parametric methods assume parametric models for the stochastic process of the underlying assets, govern the models, and derive a fair value of the option either analytically or numerically from the expected present value of the payoff its owner will receive under the risk-neutral measure. In this case, popular numerical methods for deriving the option's value include finite-difference methods and Monte Carlo simulation. In contrast, nonparametric methods do not rely on preassumed models of the underlying stochastic process, but try instead to induce the model from vast quantities of historical data.

The use of regression in option pricing usually relates to nonparametric and semiparametric methods, in which developers seek to induce the option pricing model from large quantities of historical data instead of modeling it analytically. Analysts use three common approaches to this problem. Model-free option pricing uses neural networks, genetic programming, kernel methods, nearest-neighbor, and projection pursuit regression to model the relation between the option price and all related variables. The hybrid approach starts from known approximations and uses regression methods such as neural networks and genetic programming to model the residuals and obtain a better pricing result. The risk-neutral density approach models the risk from historical data, thereafter using the estimated density to calculate the option price.

Risk management embodies the process of identifying, assessing, and developing strategies to manage the risks threatening an organization's assets or earning capacity.

The literature has grown rapidly since 1977, when Phelim Boyle[9] pioneered Monte Carlo simulation in option pricing. The basic idea of this approach is to simulate the price of the underlying asset based on a parametric model, and then use the simulation results to calculate the option's expected payoff.

Case study. A method that uses Monte Carlo simulation to calculate the option price together with extensions can reduce the execution time required for achieving acceptable precision.

In risk-neutral pricing, a fair value for an option is the present value of the option payoff at expiration under a risk-neutral random walk for the underlying asset prices. Therefore, the general approach using Monte Carlo simulation starts by finding the option's price. The term risk-neutral highlights the assumption that the current value is equal to the expected future payoff of the asset, discounted at a risk-free interest rate. Consequently, the risk-neutral random walk is basically a random walk with the drift parameter equal to the risk-free interest rate.

Using the risk-free measure, we simulate the sample paths of the underlying state variables, such as underlying asset prices and interest rates, over the relevant time. We evaluate a security's discounted cashflows on each

sample path, then average the discounted cashflows over all sample paths.

In effect, this method computes the expected value of the discounted payouts over the space of all sample paths. For example, using the Black-Scholes model[1] to simulate stock prices, Monte Carlo simulation is preferable relative to other techniques because of its flexibility, easy implementation, and ability to handle complex financial instruments. Of further benefit, its error convergence rate is often independent of the problem's dimension.

Risk management

The increasingly important topic of risk management embodies the process of identifying, assessing, and developing strategies to manage the risks threatening an organization's assets or earning capacity. These activities normally involve three basic steps:

- identification and classification of the risks that can affect a firm's business outcomes;
- measurement of the risk associated with a set of potential events that affect the firm's value in terms of their likelihood of occurring and the magnitude of expected losses; and
- formulation of the actions required to bring risks within acceptable bounds.

Normally, each risk type has its own characteristics and requires its own management mechanism. For example, market risk—the risk that an investment's value will decrease due to moves in market factors—is generally measured by value at risk, a statistical measure of downside risk that directly expresses risk in dollars or the reference currency, and is regulated by imposing capital requirements on organizations' market risk exposures.

Techniques. The wide range of risk management methods includes classification, optimization, regression, and simulation.

Classification methods play an important role in risk measurement, especially when risk is measured by one of several categories instead of a real number such as a credit rating problem,[10] which concerns the scoring of banks into the credit rating categories A+, A, A-, B+, …, E+, E, and E-.

Optimization methods frequently involve a tradeoff between risk and return, such as maintaining credit exposure within an acceptable level. In portfolio selection, analysts use computational techniques such as genetic algorithms and reinforcement learning to solve this problem when traditional optimization methods cannot.

Analysts use regression methods for modeling and forecasting future risk behavior. For example, neural networks and support vector regression are used to predict value at risk, logit/probit regression is used to estimate the probability of default, and neural networks and GARCH (generalized autoregressive conditional heteroscedasticity) models are used to forecast volatility.

Simulation is one of the most important analytical methods for risk assessment, being used frequently to estimate the value of several risk measures, such as volatility, value at risk, and default probability. The basic approach uses Monte Carlo simulation to generate numerous future scenarios, then uses the generated scenarios to compute the desired risk measurement.

Case study. The risk management case study uses classification to solve credit ratings problems.[10]

The objective is to use logistic regression, a neural network, and a support vector machine to model credit rating behavior by using publicly available financial information. The model's input includes 21 financial indicators and ratios: total assets, total liabilities, long-term debts/total invested capital, debt ratio, current ratio, times interest earned, operating profit margin, (shareholders' equity + long-term debt)/fixed assets, quick ratio, return on total assets, return on equity, operating income/received capital; net income before tax/received capital; net profit margin, earnings per share of gross profit margin, nonoperating income/sales, net income before tax/sales, cash flow from operating activities/current liabilities, cash flow from operating activities/(capital expenditures + increase in inventory + cash dividends) in the past five years, and cashflow from operating activities—cash dividends/(fixed assets + other assets + working capital).

The output is the credit rating categories of the input company. In experimental studies, a support vector machine achieved the best performance, while a neural network consistently outperformed logistic regression models.[10] In this case, the result shows that the relationship between input variables and the output variable is nonlinear since nonlinear models outperform linear ones.

A notable result of the recent financial crisis is that banks and funds seek to greatly improve their analytics and computing, and they are recruiting more science and engineering PhDs to do so. However, banks and funds struggle to attract PhDs who excel in all three key areas: programming, analytical methods, and finance. In response, departments of computer science, mathematics, and statistics are introducing more finance-related courses, and departments of finance and economics have started teaching more programming courses.

References

1. J.C. Hull, *Options, Futures, and Other Derivatives*, 7th Int'l ed., Prentice Hall, 2008.
2. H. Kitano, "Computational System Biology," *Nature*, Dec. 2002, pp. 206-209.

3. J.E. Gentle, *Computational Statistics*, Springer, 2009.
4. T. Mitchell, *Machine Learning*, McGraw Hill, 1997.
5. M. Fontana, "Simulation in Economics: Evidence on Diffusion and Communication," *J. Artificial Societies and Social Simulation*, vol. 9, no. 2, 2006, p. 8.
6. S. Mahfoud and G. Mani, "Financial Forecasting Using Genetic Algorithms," *Applied Artificial Intelligence*, vol. 10, 1996, pp. 543-565.
7. N. El Gayar et al., "An Empirical Comparison of Machine Learning Models for Time Series Forecasting," *Econometric Rev.*, to appear in 2011.
8. J. Li and E.P.K. Tsang, "Investment Decision Making Using FGP: A Case Study," *Proc. Congress on Evolutionary Computation* (CEC 99), IEEE Press, 1999, pp. 6-9.
9. P.P. Boyle, "Options: A Monte Carlo Approach," *J. Financial Economics*, May 1977, pp. 323-338.
10. Z. Huang et al., "Credit Rating Analysis with Support Vector Machines and Neural Networks: A Market Comparative Study," *Decision Support Systems*, Sept. 2004, pp. 543-558.

Chaiyakorn Yingsaeree *is a PhD student in the Department of Computer Science at University College London. He is supported by a scholarship from the National Electronics and Computer Technology Center, Thailand. Contact him at c.yingsaeree@cs.ucl.ac.uk.*

Giuseppe Nuti *is a managing director at Citadel Securities in New York. Nuti is an Honorary Senior Research Fellow in the Department of Computer Science at University College London. Contact him at guseppe.nuti@citadelsecurities.com.*

Philip Treleaven *is professor of computing at University College London and director of the Centre of Financial Computing, a partnership of UCL, LSE, and the London Business School. His research group pioneered the early automated fraud detection techniques used in finance. Treleaven's interests include computational finance and artificial intelligence. He received a PhD from the University of Manchester, UK, and is a member of IEEE and the IEEE Computer Society. Contact him at p.treleaven@ucl.ac.uk.*