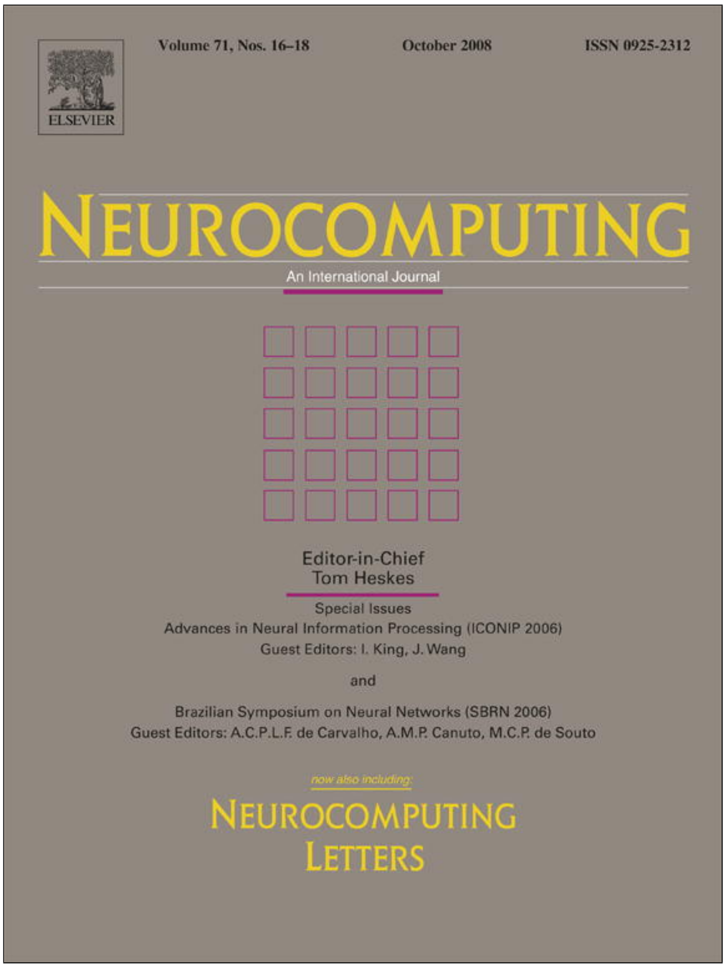


Volume 71, Nos. 16–18 October 2008 ISSN 0925-2312

ELSEVIER

NEUROCOMPUTING

An International Journal

Editor-in-Chief
Tom Heskes

Special Issues
Advances in Neural Information Processing (ICONIP 2006)
Guest Editors: I. King, J. Wang

and

Brazilian Symposium on Neural Networks (SBRN 2006)
Guest Editors: A.C.P.L.F. de Carvalho, A.M.P. Canuto, M.C.P. de Souto

now also including:

NEUROCOMPUTING
LETTERS

Neurocomputing 71 (2008) 3303–3315

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A cerebellar associative memory approach to option pricing and arbitrage trading

S.D. Teddy [a], E.M.-K. Lai [b], C. Quek [c,*]

[a] Data Mining Department, Institute for Infocomm Research (I^2R), 21 Heng Mui Keng Terrace, Singapore
[b] Institute of Information Sciences and Technology, Massey University, Wellington, New Zealand
[c] Center for Computational Intelligence, Block N4 #2A-32, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore

ARTICLE INFO

Available online 1 July 2008

Keywords:
Option pricing
PSECMAC
Cerebellum
Associative memory
Brain-inspired

ABSTRACT

Option pricing is a process to obtain the theoretical fair value of an option based on the factors affecting its price. The classical approaches to option pricing include the *Black–Scholes pricing formula* and the *binomial pricing model*. These techniques, however, employ complex and rigid statistical formulations that are not easily comprehensible to novice investors. More recently, non-parametric and computational methods of option valuation that are able to construct a model of the pricing formula from historical data have been proposed in the literature. However, most of these models functioned as black-boxes and may not be able to efficiently and accurately capture the complex market dynamics and characteristics of the option data. This paper proposes a novel brain-inspired cerebellar associative memory model for pricing American-style call options on British pound vs. US dollar currency futures. The proposed model, named PSECMAC, constitutes a local learning model that is inspired by the neurophysiological aspects of the human cerebellum. The PSECMAC-based option-pricing model is subsequently applied in a mis-priced option arbitrage trading system. Simulation results show an encouraging return on investment of 23.1% for some of the traded options.

1. Introduction

Options are financial instruments, which provide a means to manage financial risks that arise from the uncertainty of factors such as volatile interest rates, exchange rates, stock prices and commodity prices in the course of running a business. They are playing an increasingly important role in modern financial markets [13]. The buyer of an option enters into a contract with the right, but not the obligation, to purchase or sell an underlying physical or financial asset at a later date at a price agreed upon today. By using options, companies and individuals can transfer, for a price known as the *premium* (i.e. price) of the option, any undesirable financial risk to parties who either possessed the capabilities to offset such a risk (through hedging) or want to assume that risk as speculation for financial gains. Options belong to a class of financial products generally referred to as *derivative securities* whose returns are derived from those of other financial instruments, in this case, the underlying physical or financial assets for which the options are issued [12].

In an *efficient market* [18], the prices of the options being traded reflect or approximate their true economic (intrinsic) values to the investors. Classical economic models such as the *rational expectations theory* [41] and the *efficient market hypothesis* [19] assumed a stationary world in which all option investors have access to the same market data, behave rationally towards the current and historical pricing information and subsequently adopting the same trading decisions. Hence, an efficient market is always at its equilibrium. Such a notion suggested that it is a futile effort to devise a trading system to compute the fair economic value of an option and to exploit any arbitrage opportunities arising from a misalignment of the expected theoretical fair value of the option and the bid-ask price spread offered by the market. A trading market that is at equilibrium has no arbitrage opportunities.

However, in a real world financial market, information efficiency is generally far from perfect. Investors require time to learn and comprehend the inflow of information to facilitate decision-making that drives their trading actions. In addition, traders may interpret the same information differently and thus their expectations become indeterminate, unstable and possibly self-fulfilling [9]. Such a notion is also supported by modern economic models and theories such as the noisy expectations model [22] and the rational beliefs theory [36,37]. The noisy expectations model contemplated that the market *learns* while

[*] Corresponding author. Tel.: +65 6790 4926; fax: +65 6792 6559.
E-mail addresses: sdt@pmail.ntu.edu.sg (S.D. Teddy), e.lai@massey.ac.nz (E.M.-K. Lai), ashcquek@ntu.edu.sg (C. Quek).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.04.039

adapting to the incoming information, and allows for the possibility of profitable trading by using and analyzing certain information to take a position in anticipation of the price changes that will occur as the rest of the market learns about that piece of information [14]. The rational beliefs theory, on the other hand, argued that competing theories and beliefs about the future risks and the future movements of the market would result in different responses by the investors and market participants [38]. Hence, modern economic research has provided the necessary motivation and support for the need of an accurate and computationally efficient option-pricing model and the subsequent construction of a mis-priced option trading system to exploit any arbitrage opportunity to maximize trading profits.

Option pricing is defined as the process to obtain the theoretical fair economic value of an option. The price of an option is determined by a set of pricing factors such as time to expiry and the intrinsic value of the option. The conventional approach to option pricing is to construct parametric models that are based on the assumptions of continuous-time finance theory [42]. The pioneering models are the Black–Scholes (BS) formula [10] and the *binomial pricing model* [46]. This line of research has focused on the idea of creating risk-free portfolios for trading of options through dynamic hedging strategies. However, these models presumed complex and rigid statistical and probabilistic formulations about the price processes of the underlying assets from which the options prices are deduced [45]. A misspecification of the stochastic processes for the price dynamics of the underlying assets will lead to systematic pricing errors for the options linked to the assets. In addition, the theory of continuous-time stochastic processes is an essential prerequisite for continuous-time finance [42]. Such convoluted mathematical formalization, however, is not easily accessible to novice investors to develop a comprehensive understanding of the rationale behind the computed price valuations.

Non-parametric methods of option pricing based on neural networks [5,6,44,25], genetic algorithms [29] and kernel regression [1], on the other hand, are model-free approaches that have attracted significant interest in recent years. The pricing model, which is usually represented as a non-linear functional mapping between the input factors and the theoretical option price, is derived from vast quantities of historical data. However, these methods involve heuristics and therefore suffer from poor interpretability. More recently, neuro-fuzzy approaches [51] are introduced to overcome this problem. With these techniques, a set of comprehensible semantic rules can be extracted from historical trading data for rational pricing of the options.

Currently, non-parametric option-pricing methods are generally based on a global learning paradigm, in which the system attempts to use a single formulated model to generalize or fit the behaviors/characteristics of the entire set of historical pricing data. Some researchers have argued that it is difficult, if not impossible, to obtain a general and accurate global learning model [23]. Historical option-pricing data may contain complex dynamics and pricing patterns that make it hard for a global learning model to accurately generalize the underlying pricing function. In contrast, a local learning paradigm focuses on capturing only useful local information from the observed data [11]. Instead of having a single formulated model, a local learning system can be considered as a collection of locally active models, where each sub-model is learning to generalize a different subset of the training data.

In option trading, the prices of the options are determined by a set of pricing factors, such as time to expiry and the intrinsic values of the options. The complex relationship between the valuation of an option and its influencing factors may be modeled as combinatorial associations to be extracted from the historical pricing data. This motivates the use of a local associative model as a non-parametric computational method to option pricing. In this paper, a novel brain-inspired cerebellar associative memory approach to the pricing of American-style call options on the British Pound vs. US dollar currency futures is investigated. The cerebellar associative memory model, named the Pseudo Self-Evolving Cerebellar Model Arithmetic Computer (PSECMAC) [48], is employed to approximate the functional mapping between the option price and its influencing factors. The structure of the PSECMAC network is inspired by the neurophysiological properties of the human cerebellum [26], and emulates the information processing and knowledge acquisition of the cerebellar memory.

The rest of this paper is organized as follows. Section 2 describes the architecture of the PSECMAC network and outlines the structural and parameter learning process of the network. Section 3 presents an overview of the proposed cerebellar associative memory-based option-pricing model and defines the selected input factors for the pricing of the American-style currency futures option. In Section 4, the autonomous option trading system that employs the proposed option pricing model is introduced and evaluated using real-life British Pound vs. US dollar futures option trading data. Section 5 concludes this paper.

2. The PSECMAC network

The cerebellum constitutes a part of the human brain that is important for motor control and a number of cognitive functions [40], including motor learning and memory. It possesses the capability to model highly non-linear physical dynamics, and it is postulated to function as a movement calibrator [4] that is involved in the detection of movement error and the subsequent coordination of the appropriate skeletal responses to reduce the error. The human cerebellum functions by performing *associative mappings* between the input sensory information and the cerebellar output required for the production of temporal-dependent precise behaviors [26]. It has been computationally modeled by the Cerebellar Model Articulation Controller (CMAC) [3]. As a functional model of the human cerebellum, CMAC manifests as an associative memory network [2], where the memory cells are uniformly quantized to cover the entire input space. The CMAC network operation is characterized by the table lookup access of its memory cells. This allows for localized generalization and rapid algorithmic computation, and subsequently motivates the prevalent use of CMAC for process control and optimizations [56,35,39], modeling and control of robotic manipulators [16,28], as well as various signal processing and pattern-recognition tasks [53,24,30].

However, there are several major computational limitations associated with the CMAC network that arise from the rigidness of its computing structure. The CMAC network employs a highly regularized grid-like computing structure (i.e. equally spaced memory cells along each input dimension) that indirectly enforces the uniform quantization of a problem's input–output (I/O) mapping space. On the other hand, meaningful real-life applications are generally *heteroskedastic*, where the problems are often characterized by highly non-linear I/O trends and statistically varying data patterns. Such observations implied that specific regions of these problems' I/O associative spaces are more informative (and therefore demand a higher modeling resolution) than others. For such an application, the simplistic approach of adopting an uniformly quantized I/O mapping space for CMAC to model the problem's input–output data characteristics may not be adequate and often leads to suboptimal memory utilization.

This paper presents a brain-inspired cerebellar-based learning memory model named PSECMAC as a generic functional model of the human cerebellum and investigates its use as a non-parametric option-pricing model. PSECMAC differs from the CMAC network in *two* aspects. Firstly, the PSECMAC network employs *one* layer of network cells, but maintained the computational principles of the layered CMAC network by adopting a neighborhood activation of its computing cells to facilitate: (1) smoothing of the computed output; (2) distributed learning paradigm; and (3) activation of highly correlated computing cells in the input space. Secondly, instead of the conventional uniform partitioning of the memory cells as seen in CMAC, the PSECMAC network employs a data-driven adaptive memory quantization mechanism of its network cells.

The structure of the proposed PSECMAC associative memory network is inspired by neuroscience research as well as human behavioral studies on the cerebellar learning process, where it has been shown that significantly higher densities of the cerebellar synaptic connections are located at the frequently accessed regions of the cerebellum that are activated by repeated learning episodes [20]. This cerebellar-based experience-driven synaptic plasticity phenomenon is emulated in the PSECMAC network by employing a data-driven adaptive memory quantization scheme for the derivation of its computing structure. That is, more memory cells are assigned to model regions of the data space that contain higher densities of the training exemplars. Fig. 1 graphically illustrates the fundamental architectural difference in the organization of the memory (computing) cells to define the I/O mapping space between the proposed PSECMAC network and the CMAC model.

The proposed PSECMAC network employs a two-phased learning process, namely: structural learning and parameter tuning. See Appendix A for the technical details on the structural learning and parameter tuning phases of the PSECMAC network.

2.1. Computational process of PSECMAC

The PSECMAC network employs a *weighted Gaussian neighborhood output* (WGNO) computation process, where a set of neighborhood-bounded computing cells is activated to derive the network's output response to the given input stimulus. In this computation process, each of the neighborhood cells has a weighted degree of activation that is inversely proportional to the distance of the cell from the input stimulus point. The objective of the WGNO scheme is to minimize the influences of the input quantization errors on the computed network output. In addition, it introduces a "smoothing" effect on the PSECMAC output and enhances the generalization capability of the PSECMAC network.

Let $\mathbf{Y}_s$ denotes the computed PSECMAC network output for an input stimulus $\mathbf{X}_s = [x_{s,1}, x_{s,2}, \ldots x_{s,J}]^{\mathrm{T}}$ to the PSECMAC network. The WGNO computation process is defined as follows:

Step 1: *Determine the region of activation*. The size of the activated PSECMAC neighborhood with respect to input $\mathbf{X}_s$ is defined by $N \in [0 \cdots 1]$, a user-specified parameter that governs the relative size of the neighborhood of activated PSECMAC cells to the overall memory space. The neighborhood activation boundaries are defined on per-dimension basis such that $N = 0.2$ denotes an activation boundary of 20% relative to the ranges of the respective input dimensions. A neighborhood constant of $N = 0.2$ therefore signifies a neighborhood activation of $(0.2)^J \times 100\%$ relative to the entire input space, where J denotes the total number of input dimensions. For the input stimulus $\mathbf{X}_s$, its activation neighborhood is defined as

$$lb_{s,j} = x_{s,j} - 0.5 \cdot N \cdot range_j \tag{1}$$

$$rb_{s,j} = x_{s,j} + 0.5 \cdot N \cdot range_j, \quad j \in \{1, 2, \ldots, J\} \tag{2}$$

where $lb_{s,j}$ denotes the left activation boundary, $rb_{s,j}$ denotes the right activation boundary, and $range_j$ is the domain for the jth input dimension. Subsequently, the memory cells encapsulated within the neighborhood defined by the computed boundaries are activated in response to the input stimulus $\mathbf{X}_s$. A PSECMAC activation neighborhood is illustrated as Fig. 2.

Step 2: *Compute the Gaussian weighting function*. A Gaussian weighting factor g_k is associated with each activated PSECMAC cell to determine its contribution towards the computation of the network output. The Gaussian weighting factor is defined as

$$g_k = (1 - d_k)\mathrm{e}^{-d_k^2/2\gamma^2} \tag{3}$$

where γ is the Gaussian width constant and d_k denotes the normalized Euclidean distance from the kth activated cell to the input stimulus $\mathbf{X}_s$ (see Fig. 2). Let $\mathbf{K}_s$ be the set of activated PSECMAC cells in the computed neighborhood. Subsequently, d_k is defined as

$$d_k = \frac{\|\mathbf{Q}_k - \mathbf{X}_s\|}{\max_{k' \in \mathbf{K}_s} \|\mathbf{Q}_{k'} - \mathbf{X}_s\|} \tag{4}$$

where $\mathbf{Q}_k = [Q_{1,k}, Q_{2,k}, \ldots, Q_{J,k}]$ denotes the quantization point of cell k in the memory space.

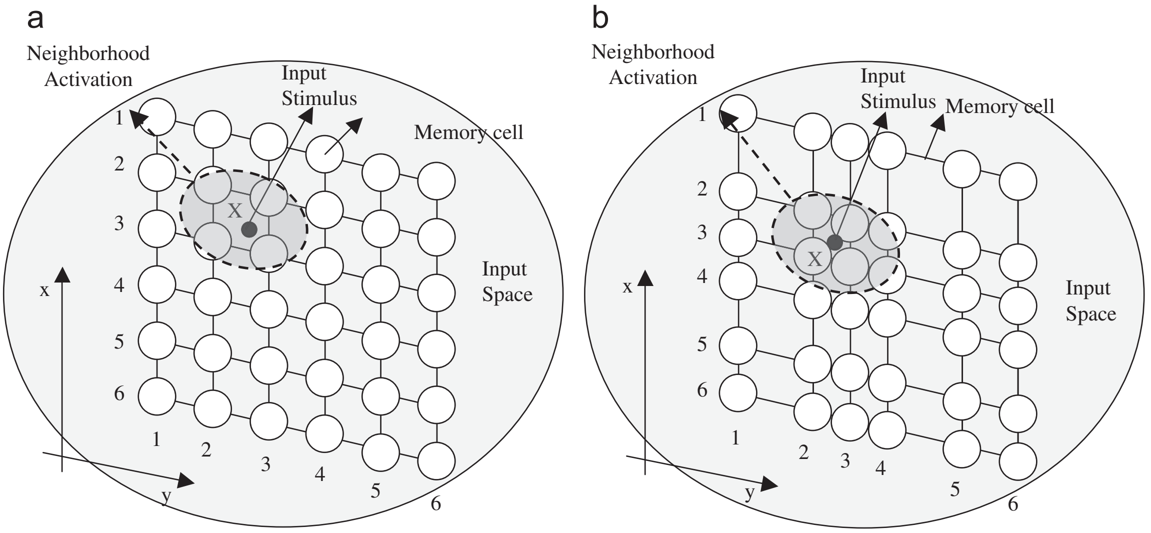


Fig. 1. Comparison of CMAC and PSECMAC memory quantization for 2D input problem. (a) CMAC and (b) PSECMAC.

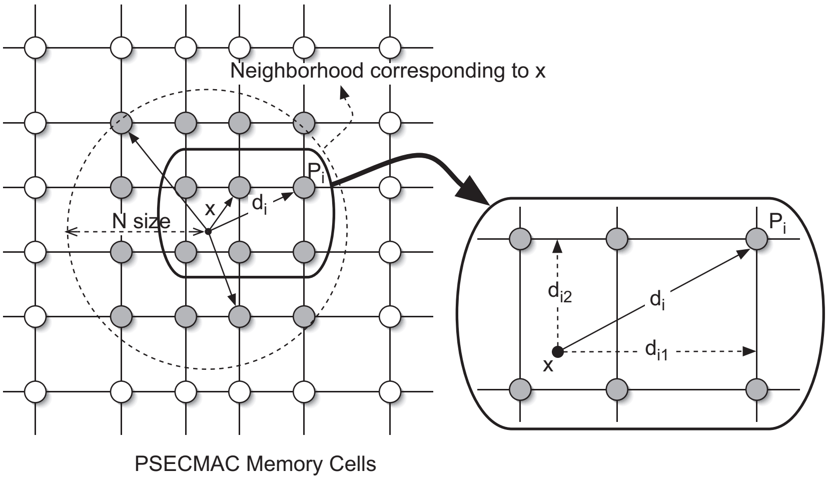


Fig. 2. An example of a 2D PSECMAC neighborhood.

Step 3: *Retrieve the PSECMAC output.* The PSECMAC network output $\mathbf{Y}_s$ is computed as a weighted linear combination of the memory contents of the activated cells such that

$$\mathbf{Y}_s = \frac{\sum_{k\in\mathbf{K}_s}(g_k \cdot \mathbf{W}(k))}{\sum_{k\in\mathbf{K}_s} g_k} \tag{5}$$

where $\mathbf{K}_s$ denotes the set of neighborhood-activated PSECMAC cells, and $\mathbf{W}(k)$ is the stored weight value(s) of the activated PSECMAC cell with index k.

3. A PSECMAC-based option-pricing model

In this paper, the use of PSECMAC as a novel non-parametric option pricing model is investigated. This section presents an overview of the proposed PSECMAC-based option-pricing model, starting with a brief discussion of the dataset used to construct the pricing model and the definition of the selected input factors considered to have an impact on the pricing of the option. In this study, the PSECMAC network is used to construct a pricing model to predict the correct valuations for American call options on the British pound (GBP) and US dollar (USD) exchange rate futures contract. The proposed pricing model is subsequently evaluated on the accuracy of its pricing decisions and the results are analyzed.

3.1. The option dataset

The data used in this study consists of the daily closing quotes of the GBP vs. USD currency futures and the daily closing bid and ask prices of American-style call options on such futures in the Chicago Mercantile Exchange (CME) [15] during the period of October 2002 to June 2003. The GBP vs. USD currency futures option in the CME have at least three unique expiration dates: current month, following month, and two months ahead. On the fourth Friday of each month, some contracts expire (as some others may be exercised earlier since they are American style options), and new ones are introduced. For any new contracts, they often have eight strike prices around the current GBP vs. USD currency futures value. If the index moves outside the current strike price range, another strike price is added for all expiration dates to bracket that index value. Thus, the strike prices reflect the path of the index during the time-to-maturity period.

3.2. Selected inputs to the option-pricing model

A study to determine the most influential factors affecting the prices of European style Swedish OMX index options [5] has identified five attributes to be very important. They are: (1) current underlying asset price (denoted as S_0); (2) time to maturity of the option (expressed in years and denoted as T); (3) the exercise price of the option, X; (4) the risk-free interest rate (generally taken as the compounded return rate on one-year Treasury Bill and denoted as r); and (5) the historical price volatility of the last 30 days (denoted as σ_{30}).

In the proposed PSECMAC-based option-pricing model, all of the above factors except the risk-free interest rate r are selected as inputs to the non-parametric pricing model. This is because the risk-free interest rate r can be assumed to be constant without change and therefore have no effect on the price of the options [54]. Thus, the American call option-pricing formula can be represented as a function of the following inputs: S_0, X, T, and σ_{30}; where S_0 is the current GBP vs. USD exchange rate futures value; X is the strike price of the option on the GBP vs. USD exchange rate futures; T is time to maturity of the option in years; and σ_{30} is the historical price volatility for the last 30 trading days. In addition, we introduce the notion of *moneyness* (or intrinsic value) of the futures option, which is computed as the difference between the current futures value S_0 and the option strike price X (i.e. $S_0 - X$), as a combined input to the PSECMAC-based option-pricing model. Thus, the pricing function f to be approximated by the PSECMAC network is

$$C_0 = f(S_0 - X, T, \sigma_{30}) \tag{6}$$

where C_0 is current option price; and $(S_0 - X)$ reflects the moneyness of the option.

3.3. Option-pricing results and analysis

In total, 792 data samples are available in the selected futures option dataset, which contains the historic pricing data for options with five different strike prices: \$158, \$160, \$162, \$166 and \$168, with 159, 158, 173, 137 and 165 data samples, respectively. The 792 data samples are subsequently partitioned into three evenly distributed sub-groups denoted as A, B and C, each containing 264 data tuples. A total of six different cross-validation sets are constructed based on the permutations of the sub-groups, as outlined in Table 1. The six CV sets are organized into two different evaluation models, namely Models 1 and 2. In Model 1, the training set is constructed using data samples from

Table 1
Simulation set-ups based on permutations of the three sub-groups A, B and C to define the training and testing sets of the proposed PSECMAC option-pricing model

Evaluation model	Configuration	Simulation	Training set	Testing set
Model 1	$\frac{1}{3}$ training and $\frac{2}{3}$ testing	I	Sub-group A	Sub-groups B and C
		II	Sub-group B	Sub-groups A and C
		III	Sub-group C	Sub-groups A and B
Model 2	$\frac{2}{3}$ training and $\frac{1}{3}$ testing	IV	Sub-groups A and B	Sub-group C
		V	Sub-groups A and C	Sub-group B
		VI	Sub-groups B and C	Sub-group A

Table 2
Performances of the proposed PSECMAC option-pricing model

Evaluation model	Simulation	Recall			Generalization		
		RMSE	NRMSE (%)	Correlation	RMSE	NRMSE (%)	Correlation
Model 1	I	0.1299	8.3	0.9956	0.2386	15.2	0.9858
	II	0.1376	8.8	0.9954	0.2727	17.4	0.9816
	III	0.1178	7.5	0.9964	0.2638	16.8	0.9847
	Average	0.1284	8.2	0.9958	0.2584	16.5	0.9840
Model 2	IV	0.1382	8.8	0.9952	0.2103	13.4	0.9889
	V	0.1404	9.0	0.9949	0.2210	14.1	0.9885
	VI	0.1353	8.4	0.9954	0.2007	12.8	0.9902
	Average	0.1380	8.8	0.9952	0.2107	13.5	0.9892

only one sub-group while the data from the remaining two sub-groups constitute the testing set. The objective of this evaluation model is to assess the generalization ability of the trained pricing system. In contrast, Model 2 employs the data samples from two sub-groups for training and aims to investigate the performances of the option-pricing system as more training samples are provided.

A PSECMAC network with a memory size of 12 cells per dimension is constructed for the option-pricing problem. A neighborhood size (N) of 0.2 and a Gaussian width constant (γ) of 0.5 have been empirically determined. Table 2 lists the *recall* (in-sample testing) and *generalization* (out-of-sample testing) performances of the PSECMAC option-pricing model for the various CV sets. RMSE denotes the root-mean-square-error between the set of computed and actual option prices; NRMSE is the normalized RMSE with reference to the average option price (in %); and PearCorr is the Pearson correlation coefficient, a statistical measure reflecting the goodness-of-fit between the approximated and actual implicit pricing functions. The average price of the GBP vs. USD currency futures option during the period of study is computed as \$1.5652. The performances of the proposed PSECMAC option-pricing model are encouraging, with an average RMSE (ARMSE) of approximately 0.13 (8.2% NRMSE) and 0.26 (16.5% NRMSE) for the recall and generalization assessments of Model 1, respectively. An average correlation of 0.98 is achieved by the PSECMAC model for the generalization evaluation (as compared to 0.99 for recall), indicating a less than 1% performance degradation as the evaluation emphasis shifts from the in-sample testing (recall) to the out-of-sample evaluation (generalization) capability of the PSECMAC pricing system.

From Table 2, one can also observe that a larger training dataset improves the generalization performance of the PSECMAC option-pricing model. The experimental results of Model 2 showed a 18% improvement ((0.2584 – 0.2107)/0.2584) in the RMSE value over that of Model 1 for generalization (out-of-sample testing). This increase in the accuracy of the PSECMAC option-pricing model can be attributed to the improvement in the network's ability to efficiently capture the price dynamics and the valuation principles of the futures option with respect to the underlying pricing factors as the number of training instances increases. A larger training dataset results in a more comprehensive training of the entire PSECMAC associative memory surface and this increases the generalization ability of the network towards the hold-out test samples.

As benchmarks, the set of option-pricing simulations is repeated using various well-established non-parametric approximators. The benchmarking models studied in this paper are: the basic CMAC network [3]; three neuro-fuzzy systems, namely: (1) the Fuzzy CMAC with Yager Inference Scheme (FCMAC-Yager) [47]; (2) the Generic Self-Organizing Fuzzy Neural Network (GensoFNN) [50]; and (3) the Rough Set-Based Pseudo-Outer-Product Fuzzy Neural Network (RSPOP) [7]; as well as the classical machine learning models such as the radial basis function (RBF) network [49] and the multi-layered perceptron (MLP). The parameters for the FCMAC-Yager, RSPOP and GenSoFNN [51] systems have all been empirically optimized for best performances. The network structure of the MLP, which consists of three input, eight hidden and one output nodes, respectively, has been empirically determined. The RBF network is initialized to contain 100 hidden layer nodes. In addition, the size of the CMAC network has been defined as 12 cells per dimension for a fair comparison with the proposed PSECMAC network. Table 3 summarizes the ARMSE, average NRMSE (ANRMSE) and average Pearson correlation (APC) results for the evaluation Models 1 and 2 across the different benchmarking architectures. A Performance Index (PI) is employed to combine the AvgRMSE and AvgPearCorr measures as described in

$$\mathrm{PI} = \frac{\mathrm{AvgPearCorr}}{(1 + \mathrm{AvgRMSE})} \times 100 \tag{7}$$

Table 3
Benchmarking results for various option-pricing models

Evaluation model	System	Recall				Generalization			
		ARMSE	ANRMSE (%)	APC	PI	ARMSE	ANRMSE (%)	APC	PI
Model 1	CMAC	0.0531	3.4	0.9992	94.88	0.2896	18.5	0.9792	75.93
	PSECMAC	0.1284	8.2	0.9958	88.24	0.2584	16.5	0.9840	78.19
	FCMAC-Yager	0.1924	12.3	0.9911	83.12	0.5221	33.4	0.9236	60.68
	GenSoFNN	0.1759	11.2	0.9943	84.56	0.2764	17.7	0.9839	77.08
	RSPOP	0.2562	16.4	0.9849	78.40	0.4204	26.9	0.9578	67.43
	RBF	0.1767	11.3	0.9920	84.30	0.3438	22.0	0.9701	72.19
	MLP(3-8-1)	0.1558	10.0	0.9938	85.98	0.1979	12.6	0.9903	82.67
Model 2	CMAC	0.0678	4.3	0.9988	93.54	0.2579	16.5	0.9834	78.18
	PSECMAC	0.1380	8.8	0.9952	87.45	0.2107	13.5	0.9892	81.70
	FCMAC-Yager	0.2365	15.1	0.9879	79.89	0.2829	18	0.9831	76.63
	GenSoFNN	0.1857	11.9	0.9948	83.90	0.2387	15.3	0.9908	79.99
	RSPOP	0.2938	18.8	0.9787	75.65	0.3461	22.1	0.9695	72.02
	RBF	0.2389	15.3	0.9854	79.54	0.3076	19.7	0.9758	74.63
	MLP(3-8-1)	0.1937	12.4	0.9905	82.98	0.2087	13.3	0.9891	81.83

Table 4
Performances of the Black–Scholes option-pricing model

Evaluation model	Simulation	Recall			Generalization		
		RMSE	NRMSE (%)	Correlation	RMSE	NRMSE (%)	Correlation
Model 1	I	1.3670	87.3	0.8289	1.2777	81.6	0.8341
	II	1.3070	83.5	0.8327	1.3087	83.6	0.8314
	III	1.2477	79.7	0.8360	1.3374	85.4	0.8302
	Average	1.3072	83.5	0.8325	1.3079	83.6	0.8319
Model 2	IV	1.3374	85.4	0.8302	1.2477	79.7	0.8360
	V	1.2777	81.6	0.8341	1.3670	87.3	0.8289
	VI	1.3087	83.6	0.8314	1.3070	83.5	0.8327
	Average	1.3079	83.6	0.8319	1.3072	83.5	0.8325

All parameters are derived directly from the spot market.

such that a higher PI value corresponds to a better pricing performance.

From Table 3, one can observe that the MLP network possesses the most accurate generalization pricing decisions as compared to the other benchmarked systems. However, it is a black-box model as its complex synaptic weight structure is hardly human interpretable. There is no mechanism to explain the pricing decisions of the MLP network. Moreover, the network structure of the MLP network has to be empirically determined. In contrast, the neuro-fuzzy systems (i.e. the GenSoFNN, RSPOP and FCMAC-Yager networks) offer interpretable semantic rules that explain the pricing decisions of the futures option but at the expense of lower pricing accuracy. The proposed PSECMAC network, on the other hand, managed to achieve comparable generalization pricing performances to those of the MLP network. The pricing decisions of the proposed PSECMAC option-pricing model outperformed the benchmarked RBF network and the neuro-fuzzy systems for both recall and generalization evaluation. Table 3 also demonstrated that the PSECMAC network outperforms its CMAC counterpart for the generalization evaluation. Specifically, the multi-resolution structure of the PSECMAC network yields (on average) a 3.7% improvement in the generalization PI value over the uniformly quantized CMAC of the same network size.

The recall performances of the PSECMAC network, however, were slightly inferior to the CMAC network. This is possibly due to the static uniform memory quantization of the CMAC network that produces a computing structure that is highly optimized for the training set. The PSECMAC non-uniform quantization procedure, on the other hand, sought to obtain an efficient characterization of a given problem's input–output mappings. This is achieved by allocating the predefined number of available memory cells in a non-linear manner based on the distribution of the training data. The effective non-linear memory quantization of PSECMAC allows for a better description of a given problem's characteristic surface to address new/unseen test data. Thus, the PSECMAC network is able to achieve a much improved generalization performance over the CMAC pricing model despite a small degradation in the recall performance. The associative structure of the PSECMAC model also facilitates the extraction of discrete pricing rules. For instance, "IF *time-to-maturity* is between 0 and 0.04 years, *volatility* is between 5.08 and 5.28 and *moneyness* is between \$5.03 and \$7.98 THEN the Option-Price (on average) is \$9.4" is a representative discrete rule extracted from the PSECMAC option-pricing model that expresses the knowledge acquired from the training data. Such rules also enhance human comprehension of the pricing dynamics of the traded options.

Subsequently, the accuracy of the non-parametric PSECMAC option-pricing model is also benchmarked against the pricing performance of the parametric BS option-pricing formula [10]. Table 4 tabulates the pricing performances of the BS option-pricing model based on the RMSE, NRMSE and the Pearson correlation coefficient values for the six CV groups. All of the parameters required by the BS pricing formula are derived from the spot market. Note that since there is no training for the BS model, the simulation results of Models 1 and 2 in Table 4 are mirrored of each other. From Table 4, one can observe that the BS pricing performances are inferior to those of the non-parametric models outlined in Table 3. Parametric option-pricing model such as the BS formula are generally derived via the continuous-time finance theory framework [42], and they often presumed complex and rigid statistical and probabilistic models about the price processes of the underlying assets to deduce the option prices. A misspecification of the price dynamics of the underlying assets will lead to systematic pricing errors that will be detrimental to the pricing performances of the parametric model. Table 4 has clearly demonstrated that the BS option-pricing model suffers from *systematic pricing errors* for the option dataset used in this study.

4. A cerebellar associative memory approach to arbitrage trading

This section introduces a mis-priced option *arbitrage* trading system, where the PSECMAC option-pricing model is employed to detect any misalignments between the market spot value and the theoretical valuation of an American call option on the GBP vs. USD currency futures. When such mis-pricing occur, potential arbitrage trading opportunities on the options are created and investors can exploit these opportunities to derive risk-free profits.

4.1. Arbitrage

An arbitrage opportunity arises when the Law of One Price [13] is violated, making it possible for an investor to make a *risk-free* profit. In this paper, an arbitrage trading strategy known as the *delta hedge trading strategy* (DHTS) [13] is employed in the proposed PSECMAC-based option trading system. In the DHTS, a delta hedge ratio h is computed to determine the quantity of the underlying asset (e.g. stock) required to cover the risk of taking a naked position on the futures call option. Hence, the selling of one call option is hedged by the buying of h quantity of the underlying asset and vice versa. The hedge ratio h is computed as

$$h_t = \frac{\Delta C}{\Delta S} = \frac{(\hat{C}_{u,t+1} - \hat{C}_{d,t+1})}{(S_{u,t+1} - S_{d,t+1})} \in [0, 1] \tag{8}$$

where h_t is the hedge ratio at current time t (i.e. this trading opportunity) employed to build up a risk-free portfolio with proper ratio of call options and the underlying asset; $S_{u,t+1}$ is the price of the underlying asset at time $t+1$ (i.e. the next trading opportunity) if the price goes up; $S_{d,t+1}$ is the price of the underlying asset at time $t+1$ (i.e. the next trading opportunity) if the price goes down; ΔS is the change in value of the underlying asset due to the projected change in price S_t at time $t+1$; ΔC is the change in value of the call options due to the projected change in price of the underlying asset at time $t+1$; $\hat{C}_{u,t+1}$ is the predicted price of the call option if the value of the underlying asset is $S_{u,t+1}$ at time $t+1$; and $\hat{C}_{d,t+1}$ is the predicted price of the call option if the value of the underlying asset is $S_{d,t+1}$ at time $t+1$.

For simplicity, the price of the underlying asset (i.e. the currency futures) in this study is assumed to either go up by 0.5 unit price or go down by 0.5 unit price (i.e. $S_{u,t+1} = S_t + 0.5$ and $S_{d,t+1} = S_t - 0.5$) such that the variable ΔS in Eq. (8) evaluates to unity. That is, there is only a unit change in the price of the underlying asset from time t to time $t+1$. Hence, Eq. (8) can be reduced to

$$\begin{aligned} h_t &= \frac{(\hat{C}_{u,t+1} - \hat{C}_{d,t+1})}{(S_{u,t+1} - S_{d,t+1})} = \frac{(\hat{C}_{u,t+1} - \hat{C}_{d,t+1})}{(S_t + 0.5 - (S_t - 0.5))} \\ &= (\hat{C}_{u,t+1} - \hat{C}_{d,t+1}) = \Delta C \end{aligned} \tag{9}$$

Thus, the hedge ratio of the portfolio at current time t is computed as the difference in the predicted prices of the call option at time $t+1$. Note that Eq. (9) also gives the expected change in the prices of the call option for every unit change in the spot price of the underlying currency futures. This is regarded as a measure of the sensitivity of the option price to the valuation of the underlying asset.

4.2. Trading strategy

Based on the DHTS discussed in the last section, the PSECMAC-based option trading system is implemented. The general framework of the trading system proposed in this paper is a modified version of the generic trading decision model found in [21], and is illustrated in Fig. 3. With respect to Fig. 3, the PSECMAC option pricing model is first constructed via supervised learning using historic pricing information derived from the option and currency futures markets. That is, PSECMAC is employed to approximate the implicit pricing function (see Eq. (6)) from the observed market valuations. The inputs and output to the PSECMAC option-pricing model during the training phase are moneyness $(S_0 - X)$, time to maturity T (in calendar year), past 30 days price volatility of the GBP vs. USD currency futures τ_{30}, and the known closing prices of the American call options C_{Closed}, respectively. The performance evaluation of this option-pricing model has been presented in Section 3. During the simulation period to evaluate the trading performance of the PSECMAC option trading system, only the required inputs (i.e. moneyness $(S_0 - X)$, time T and volatility τ_{30}) are presented to the PSECMAC pricing model to

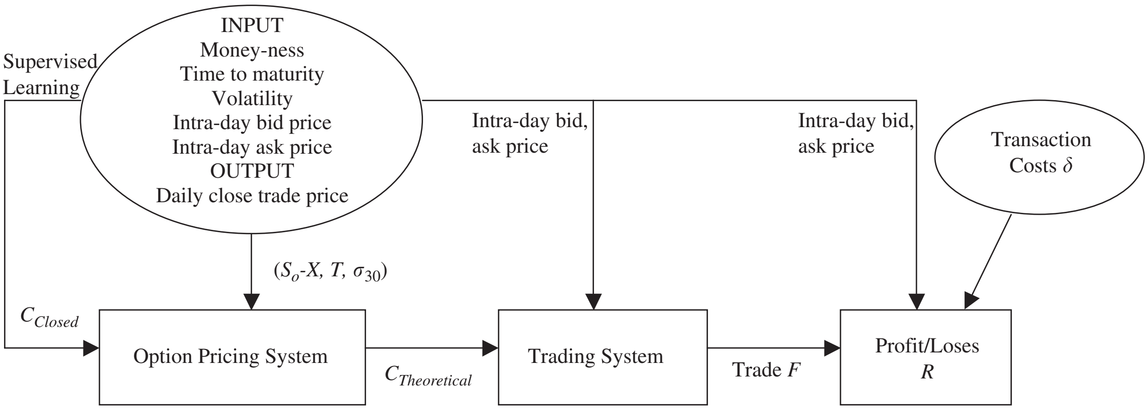


Fig. 3. General framework of the proposed mis-priced option arbitrage trading system.

compute the theoretical fair valuations (i.e. $C_{\text{Theoretical}}$) of the American call options. The computed $C_{\text{Theoretical}}$ and the spot bid-ask price spread provided by the option market are subsequently fed into the trading system to detect any occurrence of arbitrage opportunity and to generate the required trading decision F. After the trades have been executed, the profits or losses (i.e. returns R) are computed and reported at the end of the simulation period.

The format of the training set is as described in Section 3. Historic options data with strike prices of $158, $160, $162, $166 and $168, respectively, from October 2002 to February 2003 is used to train the PSECMAC-based option-pricing model. The test set contains out-of-sample data consisting of the intra-day bid and ask prices of the American options with strike prices of $158, $159, $160, $164 and $170, respectively, from January 2003 to June 2003. The trading algorithm is summarized as follows:

1. The proposed trading system takes in the theoretical option value $C_{\text{Theoretical}}$ computed by the PSECMAC-based option-pricing model and subsequently compares it to the spot bid-ask prices of the option.
2. If the predicted theoretical option value $C_{\text{Theoretical}}$ falls out of the bid-ask spread range, the trading system assumes a mis-priced arbitrage opportunity as being detected.
3. The trading system would take up trading positions according to the following trading strategy:
   (a) Evaluate if the call option is over-priced or under-priced using Eq. (10).

$$\text{Call option} = \begin{cases} \text{Over-priced} \\ \quad \text{if } C_{\text{Theoretical}} < \text{option bid-price at time } t \\ \text{Under-priced} \\ \quad \text{if } C_{\text{Theoretical}} > \text{option ask-price at time } t \end{cases} \tag{10}$$

   (b) If the call option is over-priced, short sell the call option and hedge the risk by buying in h_t quantity of the underlying asset, i.e. the GBP vs. USD currency futures. The hedge ratio h_t is computed using Eq. (9). Else, if the call option is under-priced, buy in the call option and short sell h_t quantity of the GBP vs. USD futures to hedge the risk.
4. If the trading system already possessed a portfolio (i.e. has either a long or short open position on the call option with the appropriate ratio of hedged futures), it would continuously check whether the mis-priced option has pulled back into the option bid-ask spread range. If it is the case, the trading system closes all the outstanding position immediately; else, it continues to hedge the portfolio by computing a new hedge ratio h_{t+1} and adjusting the portfolio composition.

4.3. *Trading results and analysis*

The proposed PSECMAC-based trading system is evaluated by observing its arbitrage performances using real-life GBP vs. USD currency futures option with various strike prices. To simplify the simulation setup, transaction costs are omitted here. The results are tabulated in Table 5. The "total capital outlay" column denotes the overall amount of investment made on the sales and purchases of the respective options and futures in the hedging exercises, while "return on investment" (ROI) denotes the profit earned from the trading endeavors. Table 5 is analyzed as follows. For the trading of options with a strike price of $158, there are 26 under-priced (UO) and 19 over-priced (OO) arbitrage opportunities, respectively, detected by the PSECMAC-based arbitrage trading system. The total capital outlay is $143,300 on performing the arbitrage trading activities (i.e. short-sell (buy-in) options and buy-in (short-sell) futures and the closing of the trading positions when the mis-aligned option price pulls back within the bid-ask price spread). Eventual absolute ROI is $7964.80, thus giving an effective rate of return of 5.56%.

As shown in Table 5, the PSECMAC-based trading system has demonstrated fairly high returns for investment, with an average ROI of around 12.8% across all the five options. The simulation results have also demonstrated that the PSECMAC trading system performed well and returned a high effective rate of return of 23.15% and 21.14% for the traded options with strike prices of $164 and $170, respectively. Such an observation can be explained by a price plot of the GBP vs. USD futures as shown in Fig. 4. Fig. 4 depicts a plot of the trading prices of the GBP vs. USD currency futures during the period of the simulation, together with two straight lines illustrating the options with strike prices of $158

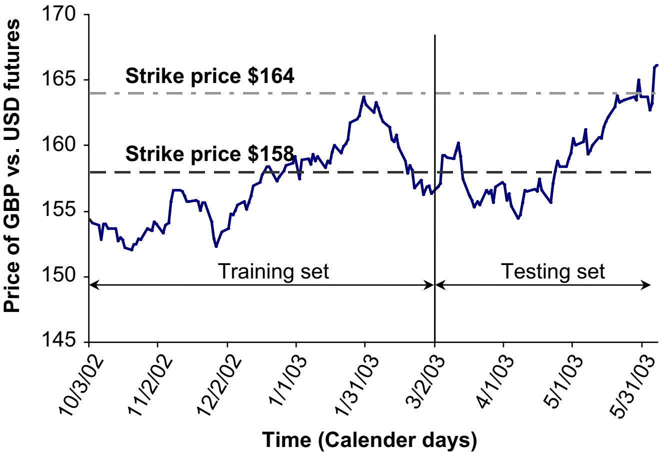


Fig. 4. Price plot of the GBP vs. USD futures for the period of the study.

Table 5
Arbitrage performances of the proposed PSECMAC-based option trading system

Option strike price X ($)	Sim period (days)	Num of UO transaction	Num of OO transaction	Total capital outlay ($)	Absolute ROI ($)	Percentage ROI (%)
158	156	26	19	143300	7964.80	5.56
159	61	7	15	50940	4228.60	8.3
160	65	0	17	30820	1809.30	5.87
164	97	17	10	20560	4759.60	23.15
170	94	10	12	5560	1175.20	21.14
Average ROI (%)						12.80

Note: UO is option under-priced arbitrage opportunity; OO is option over-priced arbitrage opportunity; and ROI denotes the return on investment.

and $164, respectively. As one may observe, during the testing (evaluation) phase of the PSECMAC trading system, the currency futures are trading at a price generally (for a majority of the evaluation period) above the support price of $158 and below $164. Hence, the options with an exercise price of $158, $159 and $160 are having high moneyness value (i.e. the options are *in-the-money*). The high moneyness of these options, in turn, reduce the profitability of the arbitrage opportunities on these options. On the other hand, more profitable arbitrage opportunities exist for the options with an exercise price of $164 and $170. This is due to the fluctuations and the continued upward trend of the futures price, which would likely diversify the market opinions and the trading decisions of the option investors. This leads to the execution of many profitable arbitrage trading by the PSECMAC option trading system as shown by the results in Table 5.

The trading performance of the proposed PSECMAC-based arbitrage trading system was subsequently benchmarked against a baseline BS-based option trading system as well as two trading strategies based on a simple buy-and-hold trading scenario. The BS-based option trading system is constructed by replacing the PSECMAC option-pricing model in the trading framework of Fig. 3. A buy-and-hold trading system, on the other hand, tries to emulate the behavior of a buy-and-hold option trader, who simply buys in the options at the start of the simulation and squares-off his position at the end of the simulation period. Two buy-and-hold trading strategies are examined in this study: (1) buy-and-hold strategy (BHS); and (2) buy-and-hold strategy with hedging (BHSH). The only difference between these two strategies is the use of hedging in BHSH to offset the risks involved in buying and adopting an open position on the options. The benchmarking results are tabulated in Table 6. All the benchmarked trading systems square-off their position at the end of the simulation period.

From Table 6, one can observe that the BS arbitrage trading system reported the poorest trading performances as compared to the rest of the benchmarked systems. The inaccuracy of the computed fair valuations of the traded options by the BS arbitrage trading system resulted in a negative overall ROI of −14.13%. This further demonstrated the significant impact of the systematic pricing errors on the performance of the BS parametric option-pricing model. The BHS trading system, on the other hand, achieved the best trading performances with an overall ROI of 207.36%. However, BHS employs a risky trading strategy due to the fact that the trader does not hedge his trading positions and therefore stands to lose all his capital if the market is not in his favor. In this study, the high ROI of the BHS trading system may be attributed to the continued upward trend of the futures price during the trading period (see Fig. 4). This leads to a major appreciation in the values of the options with lower strike prices (e.g. options with a strike price of $158, $159 and $160), as reflected in the high ROI values of these traded options. Conversely, options with higher strike prices result in a lower or even negative ROI, as in the case of the options with a strike price of $170. This clearly demonstrated the speculative and hazardous nature of this trading strategy. The BHSH trading system, on the other hand, is explored in this study to provide a fairer comparison to the PSECMAC and BS-based trading systems that perform hedging. The BHSH trading system buys options at the start of the simulation and hedge its trading position by short-selling the currency futures, and finally squares-off its trading position at the end of the simulation period with no further trading performed in-between the two trades. Although BHSH hedges its trading position at the start of the simulation, this trading strategy is almost as risky as the BHS trading strategy. This is due to the fact that BSHS does not perform dynamic hedging of the traded options. Therefore, any big fluctuations of the futures price in between the trading period may lead to a substantial financial loss. Similar to the BHS system, the high ROI values achieved by the BHSH trading system (ROI = 37.62%) in this study is because of the general up-trend of the futures price. However, in the real world trading market, this scenario cannot be assumed to be true as the general market is festered with random fluctuations.

Meanwhile, the PSECMAC-based option trading system outperformed its BS-based counterpart by generating an overall ROI of 7.94%. Although this ROI value is less than those of the BHS and BHSH systems, the PSECMAC-based option trading system produces this return via a relatively risk-free investment portfolio. This is demonstrated by the results in Table 6, where the PSECMAC-based option trading system achieved a positive return of investment on all the five traded options. This performance is in stark contrast to the two buy-and-hold trading strategies, which obtained negative returns on some of the traded options even in a favorable market condition such as in this study. In addition, the rate of return achieved by the PSECMAC option trading system is deemed to be highly encouraging given the risk-free nature of the investment portfolio constructed and when compared against other risk-free investments available during the same time period. For example, according to the US Federal Reserve Board, the 3-months compounding interest rate of the US Treasury Bill is 0.93% on 30th September 2003, while the 3-month fixed deposit interest rate in Singapore is only 0.25% on 3rd October 2003 according to the financial data provided by the Development Bank of Singapore (DBS).

Table 6
Comparison of arbitrage performances by the proposed PSECMAC-based arbitrage trading system (PSECMAC), the BS-based arbitrage trading system (Black–Scholes), simple buy-and-hold strategy (BHS) and buy-and-hold strategy with hedging (BHSH)

Option strike price ($)	PSECMAC		Black–Scholes		BHS		BHSH	
	Total capital outlay ($)	Absolute ROI ($)	Total capital outlay ($)	Absolute ROI ($)	Total capital outlay ($)	Absolute ROI ($)	Total capital outlay ($)	Absolute ROI ($)
158	143 300	7964.80	5900	−4820	1740	6400	1740	4169.70
159	50 940	4228.60	8300	−3300	1840	5300	1840	974.26
160	30 820	1809.30	4100	1620	1760	3380	1760	−22.13
164	20 560	4759.60	7300	3100	1600	540	1600	−1286
170	5560	1175.20	1720	−460	400	−400	400	−1074.30
Total	251 180	19937.5	27 320	−3860	7340	15 220	7340	2761.53
ROI (%)		7.94		−14.13		207.36		37.62

5. Conclusions

This paper proposes the use of a brain-inspired cerebellar associative learning memory structure named PSECMAC to perform non-parametric option pricing of American style call options on the British pound (GBP) vs. US dollar (USD) currency futures. The PSECMAC-based option-pricing system constitutes a local learning approach to the approximation of the associative characteristics between the option price and its influencing factors. Evaluation results have demonstrated that the modeling capabilities of the proposed pricing system exceed those of the neural fuzzy system-based models, as well as the well established CMAC network. The associative structure of the PSECMAC network also enables discrete pricing rules to be extracted from the pricing system. Subsequently, the PSECMAC-based option-pricing model is employed in a mis-priced option arbitrage trading system. In this study, the performance of the PSECMAC option trading system is benchmarked against the Black–Scholes option trading system as well as two other systems based on the simple buy-and-hold trading strategy. Simulation results on the various options with different strike prices demonstrated that the PSECMAC-based mis-priced arbitrage trading system is able to construct risk-free investment portfolios with an encouraging rate of return on investment. Future studies will attempt to incorporate other external factors such as transaction costs, as well as to extend the PSECMAC-based arbitrage trading system to support the dynamic generation of trading rules.

Appendix A. The PSECMAC network learning process

The proposed PSECMAC network employs a two-phased learning process, namely: structural learning and parameter tuning. The objective of the structural learning phase is to create the PSECMAC network's associative structure by computing the quantization decision functions for each input dimension. Subsequently, the input to output associative information of the training data samples are learnt by adapting the memory contents of the PSECMAC network in the parameter tuning phase. This appendix describes the operations involved in the structural learning and parameter tuning process of the PSECMAC network.

A.1. Structural learning of PSECMAC

The initial step in the PSECMAC structural learning phase is to identify the regions of the I/O space with high data densities. Subsequently, more memory cells (i.e. a finer network output granularity) are assigned to these regions to emulate the experience-driven dendritic aborization phenomenon observed in the cerebellar learning process during skill acquisition [31–33,20]. Analogical to the repeated exposures of the learning episodes during skill acquisition, these identified regions of the I/O space contain a large amount of training data points that coexisted in close proximity. The PSECMAC memory allocation and non-uniform quantization process is performed individually for each input dimension and consisted of several steps: (1) the identification of the data density clusters; (2) the allocation of the PSECMAC memory cells based on the computed density profile; and (3) the derivation of the respective PSECMAC quantization decision functions.

A.1.1. The computation of data density clusters

In the proposed PSECMAC network, significant data clusters supporting the inherent organization of the training dataset are first identified via the Pseudo Self-Evolving Cerebellar (PSEC) [8] clustering algorithm through an analysis of the density distribution of the training data points along each input dimension. The PSEC algorithm is a density-based clustering algorithm which synergizes the merits of the incremental learning procedure of the *learning vector quantization* (LVQ) [34] technique with the effectiveness of the density-based partitioning method of the DBSCAN algorithm [17]. This clustering algorithm is inspired by the biological development of the human brain where neural cell death plays an integral part in the refinement process of the brain's neuronal organization [8]. Neurophysiological studies have established that there are two overlapping stages in the development of the human brain [27]. The first stage of this development process encompasses the formation of the basic architecture of the brain system, in which coarse connection patterns emerge as a result of the genesis of the brain cells during prenatal development. Subsequently, in the second stage of the brain's development, the initial architecture is refined and extraneous synaptic connections are pruned throughout an individual's life-span via exposures to various activity-dependent experiences. These two stages of the human brain adaptation process are functionally emulated by the PSEC clustering algorithm.

The proposed PSECMAC network employs a modified PSEC (MPSEC) clustering algorithm to identify the centers of the density clusters along each input dimension of the training data space. A *density cluster* is defined as a cluster identified from the data density profile computed with the MPSEC algorithm. Each density cluster is associated with a *cluster center*, which denotes the point of highest data density in the cluster. The MPSEC algorithm commences with an initial set of regularly spaced density clusters, with the mid points of these initial density clusters defined as the respective cluster centers. This initial set of clusters is incrementally *evolved* to capture the data density profile along each input dimension to derive a final set of density clusters. The LVQ iterative algorithm is subsequently employed to refine the positions of the cluster centers in this final set of density clusters.

Let J and L denotes the total number of input and output dimensions for a given dataset, respectively. Assume that a training dataset of $\mathbf{U}=\{(\mathbf{X}_1,\hat{\mathbf{Y}}_1),(\mathbf{X}_2,\hat{\mathbf{Y}}_2),\ldots,(\mathbf{X}_s,\hat{\mathbf{Y}}_s),\ldots,(\mathbf{X}_S,\hat{\mathbf{Y}}_S)\}$ is used to train the PSECMAC network, where $\mathbf{X}_s=[x_{s,1}\ x_{s,2}\ \cdots\ x_{s,J}]^{\mathrm{T}}$ denotes the sth input training vector, and $\hat{\mathbf{Y}}_s=[\hat{y}_{s,1}\ \hat{y}_{s,2}\ \cdots\ \hat{y}_{s,L}]^{\mathrm{T}}$ denotes the corresponding expected output target vector of the PSECMAC network. Let τ denotes the clustering iteration in MPSEC and $\mathbf{C}_j^{(\tau)}=\{C_{j,1}^{(\tau)},C_{j,2}^{(\tau)},\ldots,C_{j,n}^{(\tau)},\ldots,C_{j,n_{C,j}^{(\tau)}}^{(\tau)}\}$ denotes the set of density clusters along the jth input dimension at the τth iteration, where $n_{C,j}^{(\tau)}$is the corresponding total number of density clusters. Let $\mathbf{C}_j^{(-1)}$ denotes the initial set of density clusters for the MPSEC algorithm and $n_{C,j}^{(-1)}$ be the number of these regularly spaced density clusters along the jth input dimension. For each input dimension $j\in\{1\cdots J\}$, the MPSEC clustering algorithm is briefly outlined as follows:

Step 1: *Initialize the clustering parameters.* An initial number of density clusters $n_{C,j}^{(-1)}$, together with a pseudo-potential threshold β, a clustering termination criterion ε and the LVQ learning constant α_c are predefined prior to the start of MPSEC clustering iteration.

Step 2: *Construct the initial set of density clusters.* The initial set of density clusters $\mathbf{C}_j^{(-1)}$ is subsequently constructed with $n_{C,j}^{(-1)}$ regularly spaced clusters such that $\mathbf{C}_j^{(-1)}=\{C_{j,1}^{(-1)},C_{j,2}^{(-1)},\ldots,C_{j,n}^{(-1)},\ldots,C_{j,n_{C,j}^{(-1)}}^{(-1)}\}$. Each density cluster $C_{j,n}^{(\tau)}$ is associated with a cluster center $P_{j,n}^{(\tau)}$ and a density value $V_{j,n}^{(\tau)}$. In the initial set of density clusters $\mathbf{C}_j^{(-1)}$, the cluster center $P_{j,n}^{(-1)}$ is assigned to be the mid-point of the corresponding density cluster $C_{j,n}^{(-1)}$ and the density value $V_{j,n}^{(-1)}$ is initialized to zero. This step emulates the formation of the initial brain system, in which extraneous connection patterns emerge as a result of the overproduction of neurons during the prenatal brain development phase.

Step 3: *Compute the initial cluster density values.* MPSEC performs structural learning by executing a one-pass learning of the density values $V_{j,n}^{(-1)}$ to obtain a density distribution of the training data along the jth input dimension.

Step 4: *Evolve the initial set of density cluster.* For each input dimension, the initial set of density clusters $\mathbf{C}_j^{(-1)}$ is evolved to capture the inherent data density profile by identifying all the local maxima in the set of computed density values $V_{j,n}^{(-1)}$. This step emulates the competitive neuronal selection process in human brain development, whereby neurons with high tropic factors are identified as the winning neurons and the remaining extraneous neurons are pruned to create a more refined structure of synaptic connections. The clusters in the initial set of density clusters $\mathbf{C}_j^{(-1)}$ whose density values form prominent convex density peaks in the computed density distribution of Step 3 are included in the new set of density clusters $\mathbf{C}_j^{(0)}$. The rest of the density clusters are removed (pruned). The clusters in the new set of density clusters $\mathbf{C}_j^{(0)}$ are therefore analogous to the surviving neurons with high tropic factors in the brain neuronal selection process.

Step 5: *Incremental learning of cluster centers.* The cluster centers $P_{j,n}^{(0)}$ of the new set of density clusters $\mathbf{C}_j^{(0)}$ are subsequently refined to derive the accurate positioning of the density-induced cluster centers. The incremental learning of the cluster centers is performed iteratively using the LVQ algorithm (i.e. $\tau = \{1 \cdots \tau_{\max}\}$), resulting in the final set of density clusters $\mathbf{C}_j^{(\tau_{\text{end}})}$, where τ_{end} denotes the last LVQ iteration performed and $1 \leqslant \tau_{\text{end}} \leqslant \tau_{\max}$.

Step 6: *Compute the resultant density profile.* A one-pass learning of the density values $V_{j,n}^{(\tau_{\text{end}})}$ in the final set of density clusters $\mathbf{C}_j^{(\tau_{\text{end}})}$ is performed to derive the density values at the final cluster centers $P_{j,n}^{(\tau_{\text{end}})}$. Finally, for each density cluster $C_{j,n}^{(\tau_{\text{end}})}$ in $\mathbf{C}_j^{(\tau_{\text{end}})}$, the left and right cluster boundary $L_{j,n}^{(\tau_{\text{end}})}$ and $R_{j,n}^{(\tau_{\text{end}})}$ are defined as the mid-point between the cluster center $P_{j,n}^{(\tau_{\text{end}})}$ and the cluster centers of its corresponding left and right neighbors.

Fig. 5 illustrates the mechanism of the MPSEC clustering algorithm. Essentially, MPSEC computes a set of density-induced clusters, whose centers denote the highest density points in the respective clusters. The boundary between any two neighboring clusters is assumed to be at the mid-point of the two respective cluster centers.

A.1.2. The PSECMAC memory allocation process

In the proposed PSECMAC network, the number of memory cells allocated to a density cluster is proportional to the normalized density value of the corresponding cluster center. Let $\hat{M}_j$ denotes the total number of user predefined memory cells in the jth input dimension. Then for each density cluster $C_{j,n}^{(\tau_{\text{end}})}$ in $\mathbf{C}_j^{(\tau_{\text{end}})}$, the number of memory cells $M_{j,n}$ allocated to this cluster is computed as

$$M_{j,n} = \left\lfloor \frac{V_{j,n}^{(\tau_{\text{end}})}}{\sum_{n' \in \{1 \cdots n_{C,j}^{(\tau_{\text{end}})}\}} V_{j,n'}^{(\tau_{\text{end}})}} \right\rfloor \times \hat{M}_j \qquad (11)$$

where $M_{j,n}$ is the number of memory cells allocated to cluster $C_{j,n}^{(\tau_{\text{end}})}$ in the jth input dimension, $V_{j,n}^{(\tau_{\text{end}})}$ is the density value of $C_{j,n}^{(\tau)}$, and $n_{C,j}^{(\tau_{\text{end}})}$ denotes the total number of computed density clusters in the jth input dimension.

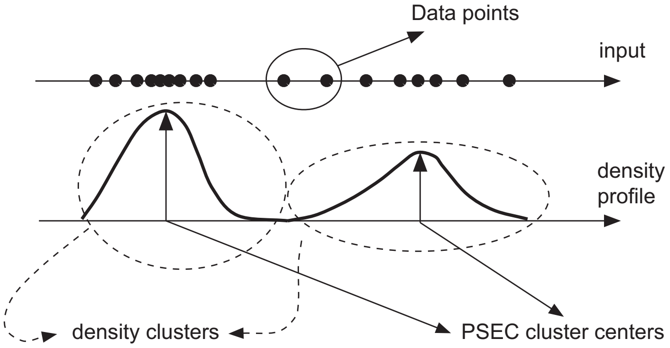


Fig. 5. A sample output of the MPSEC clustering technique.

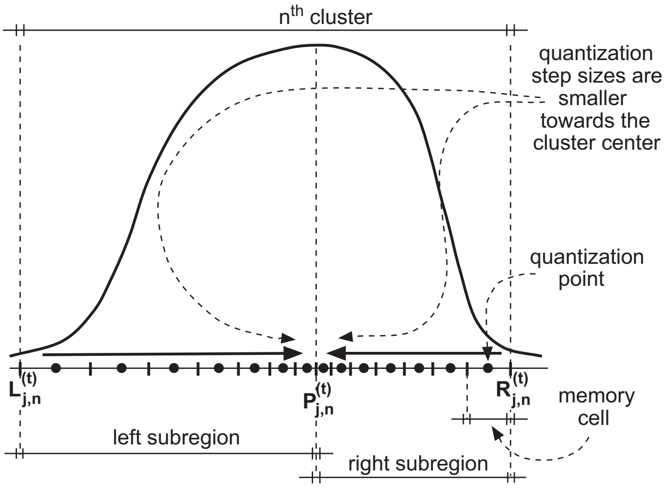


Fig. 6. Adaptive memory cell distribution in cluster $C_{j,n}^{(\tau_{\text{end}})}$.

A.1.3. The PSECMAC quantization decision functions

For the proposed PSECMAC network, a non-linear assignment scheme is introduced for the computation of the quantization decision functions to vary the quantization step sizes of the memory cells of the identified density clusters. In PSECMAC, the memory cells allocated to an arbitrary cluster $C_{j,n}^{(\tau_{\text{end}})}$ is equally distributed to the left and right side of the cluster center (i.e. the left and right subregions). In each of the two subregions, the quantization point of each memory cell is logarithmically assigned with respect to the cluster center. The quantization point of a memory cell is defined as the midpoint of the memory cell. The result of this computation is illustrated in Fig. 6, which depicts the adaptively quantized memory cells inside a cluster. Computationally, the center of each density-induced cluster constitutes the finest data granularity. As a memory cell moves away from the cluster center, its quantization step size increases in response to a lower density of the observed training data.

In this work, a logarithmic quantization technique (commonly referred to as μ-law quantization [43]) is employed to manage the distribution of the memory cells in a surviving cluster. The degree of non-linearity in the quantization step sizes of the memory cells is governed by a parameter μ. Subsequently, a quantization mapping function $\mathbf{Q}_j[\cdot] \to \{Q_{j,1}, Q_{j,2}, \ldots, Q_{j,\hat{M}_j}\}$ is constructed to define the quantization of the memory cells in the jth input dimension of the PSECMAC network, where $Q_{j,\bar{n}}$, $\bar{n} \in \{1 \cdots \hat{M}_j\}$, denotes the $\bar{n}$th quantization point. The derivation of the quantization function $\mathbf{Q}_j[\cdot]$ is described as follows:

(a) Initialize $\bar{n} = 1$ (i.e. first quantization point) and define the parameter μ for the non-linear distribution of the memory cells.

(b) For $n = 1 \cdots n_{C_j}^{(\tau_{\text{end}})}$: Let $M_{j,n}$ (computed from Eq. (11)) denotes the number of memory cells allocated to a density cluster $C_{j,n}^{(\tau_{\text{end}})}$ in the jth input dimension and k be the index to the memory cells in the density cluster $C_{j,n}^{(\tau_{\text{end}})}$. For$k = 1 \cdots M_{j,n}$, compute the quantization point for the kth memory cell in $C_{j,n}^{(\tau_{\text{end}})}$ such that:

- IF the kth memory cell is in the left subregion (i.e. $k \leqslant \lfloor M_{j,n}/2 \rfloor$) THEN:

$$step = \frac{P_{j,n}^{(\tau_{\text{end}})} - L_{j,n}^{(\tau_{\text{end}})}}{\left\lfloor \dfrac{M_{j,n}}{2} \right\rfloor} \tag{12}$$

$$pt = L_{j,n}^{(\tau_{\text{end}})} + (k - 0.5) \cdot step \tag{13}$$

$$Q_{j,\bar{n}} = L_{j,n}^{(\tau_{\text{end}})} + \left\lfloor \frac{(P_{j,n}^{(\tau_{\text{end}})} - L_{j,n}^{(\tau_{\text{end}})}) \cdot \log\left(1 + \dfrac{\mu \cdot |L_{j,n}^{(\tau_{\text{end}})} - pt|}{(P_{j,n}^{(\tau_{\text{end}})} - L_{j,n}^{(\tau_{\text{end}})})}\right)}{\log(1+\mu)} \right\rfloor \tag{14}$$

where $P_{j,n}^{(\tau_{\text{end}})}$ and $L_{j,n}^{(\tau_{\text{end}})}$ are the center and the left boundary of the density cluster $C_{j,n}^{(\tau_{\text{end}})}$, respectively. Update the index $\bar{n} = \bar{n} + 1$.

- ELSE IF $M_{j,n}$ is odd and the kth memory cell is assigned to the cluster center (i.e. $\lfloor M_{j,n}/2 \rfloor < k < \lceil M_{j,n}/2 \rceil + 1$) THEN:

$$Q_{j,\bar{n}} = P_{j,n}^{(\tau_{\text{end}})} \tag{15}$$

Update the index of the current decision point $\bar{n} = \bar{n} + 1$.

- ELSE IF the kth memory cell is in the right subregion (i.e. $k > \lfloor M_{j,n}/2 \rfloor$) THEN:

$$step = \frac{R_{j,n}^{(\tau_{\text{end}})} - P_{j,n}^{(\tau_{\text{end}})}}{\left\lfloor \dfrac{M_{j,n}}{2} \right\rfloor} \tag{16}$$

$$pt = P_{j,n}^{(\tau_{\text{end}})} + \left(k - \left\lfloor \frac{M_{j,n}}{2} \right\rfloor - 0.5\right) \cdot step \tag{17}$$

$$Q_{j,\bar{n}} = R_{j,n}^{(\tau_{\text{end}})} - \left\lfloor \frac{(R_{j,n}^{(\tau_{\text{end}})} - P_{j,n}^{(\tau_{\text{end}})}) \cdot \log\left(1 + \dfrac{\mu \cdot |pt - R_{j,n}^{(\tau_{\text{end}})}|}{(R_{j,n}^{(\tau_{\text{end}})} - P_{j,n}^{(\tau_{\text{end}})})}\right)}{\log(1+\mu)} \right\rfloor \tag{18}$$

where $R_{j,n}^{(\tau_{\text{end}})}$ is the right boundary of the density cluster $C_{j,n}^{(\tau_{\text{end}})}$. Update the index $\bar{n} = \bar{n} + 1$.

Note that the second condition (Eq. (15)) is met only when $M_{j,n}$ is odd. Otherwise, the number of allocated memory cells to the left and right subregions of cluster $C_{j,n}^{(\tau_{\text{end}})}$ is equal to $\lfloor M_{j,n}/2 \rfloor$.

After the completion of this placement process, the quantization mapping function $\mathbf{Q}_j[\cdot]$ is defined for the jth input dimension. The computed quantization decision points of each input dimension subsequently form the memory axes of the proposed PSECMAC network and are used to define its overall computing structure. The intersections of these memory axes denote the computing cells of the PSECMAC network and define the I/O associative space. The training of this PSECMAC computing structure is described in the following subsection.

A.2. Parameter tuning of PSECMAC

This section describes the parameter tuning phase of the proposed PSECMAC network. Parameter tuning is performed for the PSECMAC network to learn the mapping of the input–output associative patterns from the training data tuples. To emulate the neighborhood learning phenomenon of the human cerebellum [52,27], the PSECMAC network adopts a *weighted Gaussian neighborhood update* (WGNU) process. WGNU combines the Widrow–Hoff training algorithm [55] with the Gaussian weighting function defined in Eq. (3). The objective of this neighborhood update scheme is to distribute the effect of learning to increase the generalization capability of the PSECMAC network.

For an arbitrary input–output training data tuple $(\mathbf{X}_s, \hat{\mathbf{Y}}_s)$, the PSECMAC learning process is mathematically described as follows:

1. Compute the PSECMAC output $\mathbf{Y}_s^{(i)}$ at the ith training iteration:

$$\mathbf{Y}_s^{(i)} = \frac{\sum_{k \in \mathbf{K}_s} (g_k \cdot \mathbf{W}^{(i)}(k))}{\sum_{k \in \mathbf{K}_s} g_k} \tag{19}$$

where $\mathbf{K}_s$ is the set of activated computing cells corresponding to the input $\mathbf{X}_s$, g_k is the Gaussian weighting factor of the kth activated memory (computing) cell, $\mathbf{W}^{(i)}(k)$ denotes the memory content of the kth activated memory cell at the ith training iteration, and $\mathbf{Y}_s^{(i)}$ is the output of the PSECMAC network to the input $\mathbf{X}_s$ at the ith iteration.

2. Compute the network output error at the ith iteration:

$$\mathbf{Err}_s^{(i)} = \hat{\mathbf{Y}}_s - \mathbf{Y}_s^{(i)} \tag{20}$$

where $\mathbf{Err}_s^{(i)}$ denotes the output error of the PSECMAC network to the input $\mathbf{X}_s$ at the ith iteration, and $\hat{\mathbf{Y}}_s$ is the desired (target) output of the PSECMAC network in response to the input $\mathbf{X}_s$.

3. Update the stored network weights:

$$\mathbf{W}^{(i+1)}(k) = \mathbf{W}^{(i)}(k) + \Delta\mathbf{W}^{(i)}(k), \quad k \in \mathbf{K}_s \tag{21}$$

$$\Delta\mathbf{W}^{(i)}(k) = \alpha \underbrace{\frac{g_k}{\sum_{k' \in \mathbf{K}_s} g_{k'}} \mathbf{Err}_s^{(i)}}_{\text{local error for cell } k}, \quad k \in \mathbf{K}_s \tag{22}$$

where α is the learning constant, and $\mathbf{W}^{(i)}(k)$ denotes the content (weight) of the kth activated cell in the neighborhood $\mathbf{K}_s$ in PSECMAC in response to the input stimulus $\mathbf{X}_s$ at the ith training iteration.

The PSECMAC memory learning phase commences with the computation of the network output corresponding to the input stimulus $\mathbf{X}_s$. A learning error is computed based on the derived PSECMAC output and the target response. This error is subsequently distributed to all the activated computing (memory) cells based on the Gaussian weighting factors. The local errors are then used to update the memory contents of the activated cells.

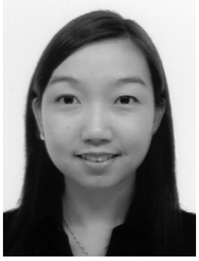
References

[1] Y. Ait-Sahalia, A.W. Lo, Nonparametric estimation of state-price densities implicit in financial asset price, LFE-1024-95, MIT-Sloan School of Management, 1995.

[2] J.S. Albus, Data storage in cerebellar model articulation controller (CMAC), J. Dyn. Syst. Meas. Control Trans. ASME (1975) 228–233.

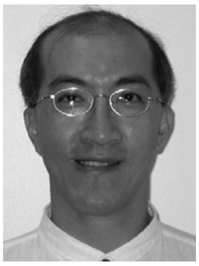
[3] J.S. Albus, A new approach to manipulator control: the cerebellar model articulation controller (CMAC), J. Dyn. Syst. Meas. Control Trans. ASME (1975) 220–227.

[4] J.S. Albus, Marr and Albus theories of the cerebellum two early models of associative memory, in: Proc. IEEE Compcon, 1989.
[5] H. Amilon, A neural network versus Black–Scholes: a comparison of pricing and hedging performances, Scandinavian Working Papers in Economics, Lund University Series, Department of Economics, Lund, Sweden, 2001.
[6] U. Anders, O. Korn, C. Schmitt, Improving the pricing of options—a neural network approach, J. Forecast. 17 (5–6) (1998) 369–388.
[7] K.K. Ang, C. Quek, RSPOP: rough set-based pseudo outer-product fuzzy rule identification algorithm, Neural Comput. 17 (1) (2005) 205–243.
[8] K.K. Ang, C. Quek, Stock trading using PSEC and RSPOP: a novel evolving rough set-based neuro-fuzzy approach, IEEE Congr. Evol. Comput. 2 (2005) 1032–1039.
[9] W.B. Arthur, Inductive reasoning and bounded rationality, Am. Econ. Assoc. Papers Proc. 84 (1994) 406–411.
[10] F. Black, N. Scholes, The pricing of options and corporate liabilities, J. Political Econ. 81 (1973) 637–659.
[11] L. Bottou, V. Vapnik, Local learning algorithms, Neural Comput. 4 (1992) 888–900.
[12] P. Boyle, F. Boyle, Derivatives: The Tools That Changed Finance, Risk Books, London, 2001.
[13] D.M. Chance, An Introduction to Derivatives & Risk Management, sixth ed., Thomson, 2004.
[14] P.Y.K. Cheng, M.I. Mah, C. Quek, Predicting the impact anticipatory action on us stock market event study using ANFIS (a neural fuzzy model), in: Proceedings of 2005 IEEE Congress on Evolutionary Computation (CEC2005), vol. 3, 2005, pp. 2669–2676.
[15] US, Chicago Mercantile Exchange [Online] ⟨http://www.cme.com⟩.
[16] S. Commuri, S. Jagannathan, F.L. Lewis, CMAC neural network control of robot manipulators, J. Robot Syst. 14 (6) (1997) 465–482.
[17] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, 1996.
[18] E.F. Fama, The behavior of stock-market prices, J. Bus. 38 (1965) 34–105.
[19] E.F. Fama, Efficient capital markets: a review of theory and empirical work, J. Finance 25 (1970) 383–417.
[20] K.D. Federmeier, J.A. Kleim, W.T. Greenough, Learning-induces multiple synapse formation in rat cerebellar cortex, Neurosci. Lett. 332 (2002) 180–184.
[21] R. Gencay, The predictability of security returns with simple trading rules, J. Empirical Finance 5 (4) (1998) 347–359.
[22] S.J. Grossman, J.E. Stiglitz, On the impossibility of informationally efficient markets, Am. Econ. Rev. 70 (393–408) (1980).
[23] K. Huang, H. Yang, I. King, M.R. Lyu, Local learning vs. global learning: an introduction to maxi–min margin machine, in: L. Wang (Ed.), Support Vector Machines, vol. 177, Springer, Berlin, 2005, pp. 113–132.
[24] K.L. Huang, S.C. Hsieh, H.C. Fu, Cascade-CMAC neural network applications on the color scanner to printer calibration, Int. Conf. Neural Networks 1 (1997) 10–15.
[25] J. Hutchinson, A. Lo, T. Poggio, A nonparametric approach to pricing and hedging derivative securities via learning networks, J. Finance 49 (1994) 851–889.
[26] E.R. Kandel, J.H. Schwartz, T.M. Jessell, Principles of Neural Science, fourth ed., McGraw-Hill, New York, 2000.
[27] E.R. Kandel, J.H. Schwartz, T.M. Jessell, Principles of Neural Science, fourth ed., McGraw-Hill, Health Professions Division, 2000.
[28] H. Kano, K. Takayama, Learning control of robotic manipulators based on neurological model CMAC, in: Proceedings of the 11th Triennial World Congress of the International Federation of Automatic Control, Tallinn, USSR, 1990, pp. 249–254.
[29] C. Keber, Option pricing with the genetic programming approach, J. Comput. Intell. Finance 7 (6) (1999) 26–36.
[30] J. Ker, C. Hsu, Y. Kuo, B. Liu, A fuzzy CMAC model for color reproduction, Fuzzy Sets and Systems 91 (1) (1997) 53–68.
[31] J.A. Kleim, E. Lussnig, E.R. Schwars, T.A. Comery, W.T. Greenough, Synaptogenesis and FOS expression in the motor cortex of the adult rat after motor skill learning, J. Neurosci. 16 (14) (1996) 4529–4535.
[32] J.A. Kleim, M.A. Pipitone, C. Czerlanis, W.T. Greenough, Structural stability within the lateral cerebellar nucleus of the rat following complex motor learning, Neurobiol. Learn. Memory 69 (1998) 290–306.
[33] J.A. Kleim, R.A. Swain, K.A. Amstrong, R.M.A. Napper, T.A. Jones, W.T. Greenough, Selective synaptic plasticity within the cerebellar cortex following complex motor skill learning, Neurobiol. Learn. Memory 69 (1998) 274–289.
[34] T. Kohonen, Self-Organization and Associative Memory, third ed., Springer, Berlin, New York, 1989.
[35] S. Ku, G.A. Larsen, S. Cetinkunt, Fast servo control for ultra-precision machining at extremely low feed rates, Mechatronics, 1998.
[36] M. Kurz, Asset prices with rational beliefs, CEPR Publication No. 375, 1994.
[37] M. Kurz, On rational belief equilibria, Econ. Theory 4 (1994) 859–876.
[38] J. Lakonishok, I. Lee, A.M. Poteshman, Investor Behavior in the Option Market, National Bureau of Economic Research, Inc., 2004.
[39] G.A. Larsen, S. Cetinkunt, A. Donmez, CMAC neural network control for high precision motion control in the presence of large friction, J. Dyn. Syst. Meas. Control 117 (1995) 415–420.
[40] F.A. Middleton, P.L. Strick, The cerebellum: an overview, Trends Cogn. Sci. 27 (9) (1998) 305–306.
[41] J.A. Muth, Rational expectations and the theory of price movements, Econometrica 29 (6) (1961) 315–335.
[42] L.T. Nielsen, Pricing and Hedging of Derivative Securities—Textbook in Continuous-time Finance Theory, Oxford University Press, Oxford, 1999.
[43] S.J. Orfanidis, Introduction to Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[44] M. Qi, G.S. Maddala, Option-pricing using artificial neural networks: the case of s&p500 index call options, Neural Networks Financial Eng. 20 (1995) 78–92.
[45] P. Radzikowski, Non-parametric methods of option pricing, in: Proceedings of Informs-Korms (Seoul 2000 Conference), 2000, pp. 474–480.
[46] R.J. Rendleman Jr., B.J. Bartter, Two-state option pricing, J. Finance 34 (1979) 1093–1110.
[47] J. Sim, W.L. Tung, C. Quek, FCMAC-Yager: a novel Yager inference scheme based fuzzy CMAC, IEEE Trans. Neural Networks 17 (6) (2006) 1394–1410.
[48] S.D. Teddy, C. Quek, E.M.-K. Lai, PSECMAC: a novel self-organizing multiresolution associative memory architecture, IEEE Trans. Neural Networks 19 (4) (2008) 689–712.
[49] The University of Waikato, WEKA 3: Data mining software in java [Online] ⟨http://www.cs.waikato.ac.nz/ml/weka/⟩.
[50] W.L. Tung, C. Quek, GenSoFNN: a generic self-organizing fuzzy neural network, IEEE Trans. Neural Networks 13 (5) (2002) 1075–1086.
[51] W.L. Tung, C. Quek, GenSo-OPATS: a brain-inspired dynamically evolving option pricing model and arbitrage trading system, in: Proceedings of the IEEE CEC 2005, vol. 3, 2005, Edinburgh, Scotland, pp. 2429–2436.
[52] J. Voogd, M. Glickstein, The anatomy of the cerebellum, Trends Cogn. Sci. 2 (9) (1998) 307–313.
[53] A. Wahab, E.C. Tan, H. Abut, HCMAC amplitude spectral subtraction for noise cancellation, in: International Conference on Neural Information Processing, 2001.
[54] R.W. Whaley, On valuing American futures options, in: D.F. DeRosa (Ed.), Currency Derivatives: Pricing Theory, Exotic Options, and Hedging Applications, 1998, pp. 147–161.
[55] B. Widrow, S.D. Stearns, Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1985.
[56] T. Yamamoto, M. Kaneda, Intelligent controller using CMACs with self-organized structure and its application for a process system, IEICE Trans. Fundam. E82-A (5) (1999) 856–860.



S.D. Teddy received the B.Eng. degree (First Class Honors) in computer engineering from Nanyang Technological University, Singapore, in 2003, where she subsequently pursued her doctorate degree at the Centre for Computational Intelligence, School of Computer Engineering. She is currently a research engineer with the Data Mining Department of the Institute for Infocomm Research, Singapore. Her current research interests include the cerebellum and its computational model, artificial neural networks, the study of brain-inspired learning memory systems, computational finance, and autonomous control of bio-physiological processes.

E. M.-K. Lai received the B.E.(Hons) and Ph.D. degrees in 1982 and 1991, respectively, from the University of Western Australia, both in electrical engineering. He is currently a faculty member of the Institute of Information Sciences and Technology, Massey University at Wellington, New Zealand. Previously he has been a faculty member of the Department of Electrical and Electronic Engineering, The University of Western Australia from 1985 to 1990, the Department of Information Engineering, the Chinese University of Hong Kong from 1990 to 1995, Edith Cowan University in Perth from 1995 to 1998 and the School of Computer Engineering, Nanyang Technological University in Singapore from 1999 to 2006. His current research interests include artificial neural networks, digital signal processing and information theory.



C. Quek received the B.Sc. degree in electrical and electronics engineering and the Ph.D. in intelligent control from Heriot-Watt University, Edinburgh, Scotland, UK, in 1986 and 1990, respectively. He is an associate professor and a member of the Centre for Computational Intelligence, formerly the Intelligent Systems Laboratory, School of Computer Engineering, Nanyang Technological University. His research interests include intelligent control, neuro-cognitive architectures, AI in education, neural networks, fuzzy systems, fuzzy rule-based systems, and genetic algorithms and brain-inspired neuro-cognitive applications in Computational Finance and Biomedical Engineering.