Available online at www.sciencedirect.com

ScienceDirect

The Journal of Finance and Data Science 5 (2019) 61−82
http://www.keaipublishing.com/en/journals/jfds/

COSMOS trader − Chaotic Neuro-oscillatory multiagent financial prediction and trading system

Raymond S.T. Lee*

Computer Science & Technology Division, BNU-HKBU United International College (UIC), Zhuhai, Guangdong, China

Received 9 November 2018; accepted 28 January 2019
Available online 6 February 2019

Abstract

Over the years, financial engineering ranging from the study of financial signals to the modelling of financial prediction is one of the most stimulating topics for both academia and financial community. Not only because of its importance in terms of financial and commercial values, but more it vitally poses a real challenge to worldwide researchers and quants owing to its highly chaotic and almost unpredictable nature.

This paper devises an innovative Chaotic Oscillatory Multi-agent-based Neuro-computing System (a.k.a. COSMOS) for worldwide financial prediction and intelligent trading. With the adoption of author's theoretical works on Lee-oscillator with profound transient-chaotic property, COSMOS effectively integrates chaotic neural oscillator technology into: 1) COSMOS Forecaster - Chaotic FFBP-based Time-series Supervised-learning agent for worldwide financial forecast and; 2) COSMOS Trader - Chaotic RBF-based Actor-Critic Reinforcement-learning agents for the optimization of trading strategies. COSMOS not only provides a fast reinforcement learning and forecast solution, more prominently it successfully resolves the massive data over-training and deadlock problems which usually imposed by traditional recurrent neural networks and RBF networks using classical sigmoid or gaussian-based activation functions.

From the implementation perspective, COSMOS is integrated with 2048-trading day time-series financial data and 39 major financial signals as input signals for the real-time prediction and intelligent agent trading of 129 worldwide financial products which consists of: 9 major cryptocurrencies, 84 forex, 19 major commodities and 17 worldwide financial indices. In terms of system performance, past 500-day average daily forecast performance of COSMOS attained less 1% forecast percentage errors and with promising results of 8−13% monthly average returns.

Keywords: Financial prediction system; Intelligent agent-based trading system; Chaotic neural networks; Actor-critic reinforcement-learning; Intelligent agents

* Corresponding author. Computer Science & Technology Division, United International College (UIC), 2000 Jintong Road, Tangjiawan, Zhuhai, Guangdong Province, China.

E-mail address: raymondshtlee@uic.edu.hk.

Peer review under responsibility of KeAi.

https://doi.org/10.1016/j.jfds.2019.01.001

1. Introduction

Over the years, financial engineering ranging from various financial signals and chart patterns study to the modelling of financial prediction and trading systems is one of the most stimulating topics for both academia and financial community. Not only because of its utmost importance in terms of financial and commercial values, but more it vitally poses a real challenge to worldwide researchers and quantitative analysts (a.k.a. "quants") throughout the world.

Conventional technical analysis and chart analysis methods probe various trading signals (e.g. K-lines, MAs, Bollinger-band index, KDJ and RSI indices) and chart patterns (e.g. major reversal and trend patterns, Golden-ratio patterns, Fibonacci patterns, Elliot-wave patterns) that believe to affect the market price trends, trading patterns and individual reasonings to determine the best time to trigger the buy or sell decisions.[5−7,24] However, these numerous trading-signals and chart-patterns are usually self-contradictory between different time-frames, let's alone with the fact that they are highly subjective to the traders' own judgment and psychological condition.

With the advance of computational capacity in the past decades, the modelling of complex financial prediction systems ranging from artificial neural networks (ANNs)[8,10] to fractal-based financial forecast systems[21] using ordinary desktop PCs and workstations is not a dream anymore.

Current research on financial prediction includes: stock prediction using Deep Neural Network (DNN) with PCA (Principal Component Analysis) by Singh & Srivastava[27]; fuzzy models by Hwang & Oh[13]; Postfix-GP (Genetic Programming) models by Dabhi & Chaudhary[11]; SVM (Support Vector Machine) and hybrid SVR (Support Vector Regression) models by Henrique et. al.[12], Nahil & Lyhyaoui[25] and Ouahilal et. al.[26]

Moreover, the popularity of free and open quantitative financial system development platforms such as MetaTrader (MT) platform[1] provides an ideal environment for worldwide researchers and quants to test their trading algorithms, strategies and financial signals with real time financial data streams which prosper the popularity of program trading, especially the HFAPT (High Frequency Algorithmic Program Trading) in the past 10 years.

From the financial market perspective, the blooming of cryptocurrency originated from bitcoin in 2009 had increased to more than 4000 cryptocurrencies in the worldwide financial market. More importantly, major international fund houses and forex trading platforms integrate 24×7 electronic trading of cryptocurrency into their forex trading platforms since 2013. Together with the flourish of HFAPT in the past 10 years, the worldwide financial markets; especially the international currency market become more volatile and unpredictable. An effective, open and reliable worldwide financial prediction, trading and advisory system is profoundly required for worldwide traders and investors (especially independent investors) than ever before.

This paper devises an innovative **C**haotic **Os**cillatory **M**ulti-agent-based Neur**o**-computing **S**ystem (a.k.a. "COSMOS") for worldwide financial prediction and intelligent trading.

With the adoption of author's theoretical works on Lee-oscillator with profound transient-chaotic property,[15] COSMOS effectively integrates chaotic neural oscillator technology into: 1) COSMOS Forecaster - Chaotic FFBP-based Supervised-learning agent for worldwide financial forecast and; 2) COSMOS Trader - Chaotic RBF-based Actor-Critic Reinforcement-learning agents for the optimization of trading strategies. COSMOS not only provides a fast reinforcement learning and forecast solution, more importantly it successfully resolves the massive data over-training and deadlock problems which usually imposed by traditional recurrent neural networks and RBF networks using classical sigmoid or gaussian-based activation functions.

From the implementation perspective, COSMOS is integrated with 2048-trading day time series financial data and 39 major financial signals as input signals for the real-time prediction and agent-based trading of 129 worldwide financial products which consists of: 9 major cryptocurrencies, 84 forex, 19 major commodities and 17 worldwide financial indices.

2. An overview of chaotic neural networks

Over decades, traditional Artificial Neural Networks (ANNs) using simple artificial neurons as constituting elements are refuted to be oversimplification to simulate real-world problems. For problems with complex and highly chaotic behaviors such as severe weather situations like rainstorms or wind-shear, or highly fluctuated real-time currency markets, there is strong evidence that neural network with the adoption of neural oscillators seems to be a more viable solution.[9,16]

In contrast with most computational intensive neural oscillators using time-continuous-based architecture, Wang[28] proposed a simple but effective time-discrete-based neural oscillator so-called "Wang-oscillator". Its bifurcation diagram provides a strong evidence to be served as a CTU (Chaotic Transfer Unit) to model complex and chaotic phenomena.

A typical Wang-oscillator consists of 3 neural elements: E, I, and W which corresponds to the Excitatory, Inhibitory and Output neurons. The neural dynamics are given by:

$$\mathrm{E}(\mathrm{t}+1) = Sig[\omega_{EE} \bullet E(t) - \omega_{EI} \bullet I(t) + S_E(t) - \xi_E] \tag{1}$$

$$\mathrm{I}(\mathrm{t}+1) = Sig[\omega_{IE} \bullet E(t) - \omega_{II} \bullet I(t) + S_I(t) - \xi_I] \tag{2}$$

$$\mathrm{W}(\mathrm{t}) = E(t) - I(t) \tag{3}$$

where S_E and S_I are the input stimulus; ω are the weights; ξ_E and ξ_I are the threshold values; the Sigmoid function Sig() is given by:

$$Sig(k) = {}^{1}\!/_{1+e^{-k}} \tag{4}$$

Fig. 1 shows the bifurcation charts of Wang-oscillator vs. Lee-oscillator. As shown in Fig. 1a, the bifurcation chart of a typical Wang-oscillator consists of "Sigmoid" and "Bifurcation" zones. The original idea is that the "Sigmoid zone" imitates the classical sigmoid function and the "Bifurcation zone" imitates the chaotic property of complex system during the sigmoid-transition period. However, there are two intrinsic problems: 1) the existence of an extra "Bifurcation zone I" destructs the continuity of the sigmoid curve in the initial period; 2) the "Bifurcation zone II" is "too chaotic" to model the chaotic-transition region in real-world problems. A more transient progressive-growth in terms of neural dynamics is needed to be achieved.

Lee-oscillator (Lee, 2006a;[17]) successfully emulates the transient-chaotic progressive growth in its neural dynamics, which helps to shed light on acting as a perfect CTU to model complex and chaotic problems.

Basically, Lee-oscillator consists of 4 neural elements: E, I, Ω and L which corresponds to the Exhibitory, Inhibitory, Input and Output neurons.

$$\mathrm{E}(\mathrm{t}+1) = Sig[e_1 \bullet E(t) - e_2 \bullet I(t) + S(t) - \xi_E] \tag{5}$$

$$\mathrm{I}(\mathrm{t}+1) = Sig[i_1 \bullet E(t) - i_2 \bullet I(t) - \xi_I] \tag{6}$$

$$\Omega(\mathrm{t}+1) = Sig[S(t)] \tag{7}$$

$$\mathrm{L}(\mathrm{t}) = [E(t) - I(t)] \bullet e^{-kS^2(t)} + \Omega(t) \tag{8}$$

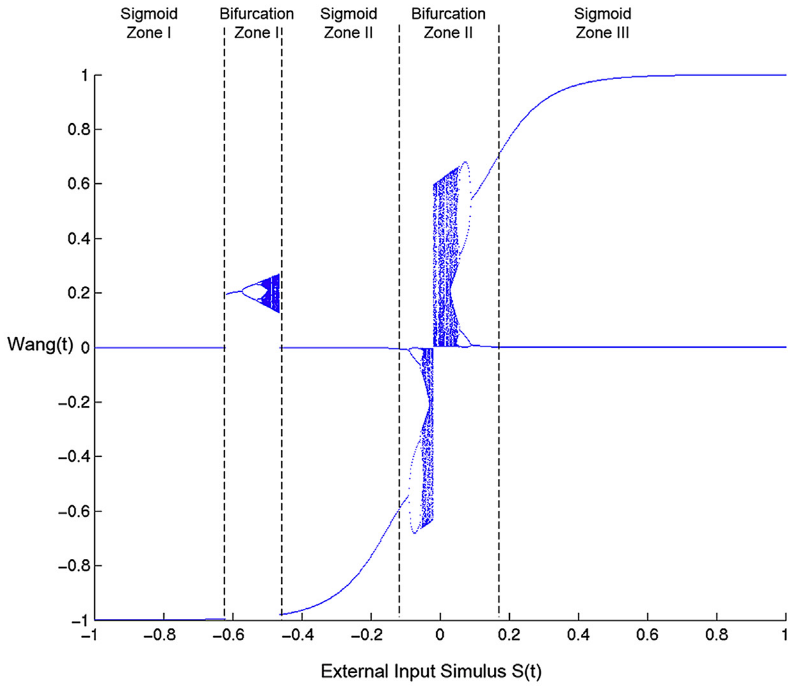
where e_1, e_2, i_1 and i_2 are the weights; ξ_E and ξ_I are the threshold values and S(t) is the external input stimulus.

As compared with the bifurcation chart of Wang-oscillator, the bifurcation chart of Lee-oscillator exhibits a perfect "sigmoid-like" function with chaotic property in the transition region (so called "bifurcation zone"). Such neural dynamics can be perfectly served as CTU to model complex and chaotic systems such as severe weather situation,[29] complex scene analysis[19] and real time financial prediction and trading agents discussed in this paper.

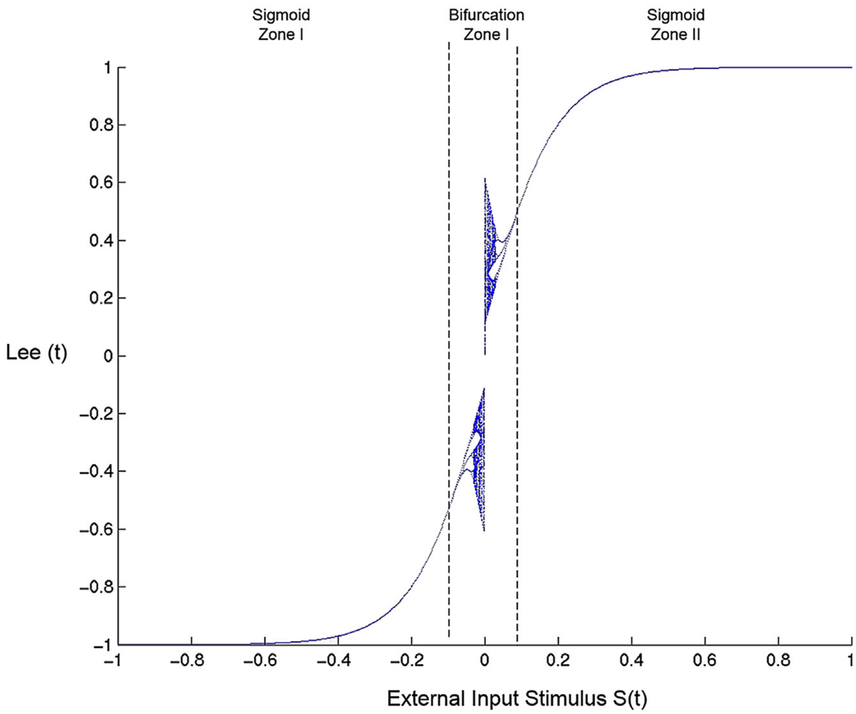
3. Chaotic FFBP-based time-series supervised-learning neural networks for financial prediction

3.1. Time series prediction using supervised-learning-based artificial neural networks

Time series prediction, ranging from weather forecast to stock prediction have been studied for over 50 years. With the improvement of computational speed, non-linear models as such Artificial Neural Networks (ANN) have proven success to tackle with these problems. Traditional ANN such as FFBPN (Feedforward Back-Propagation Neural Network) is a typical kind of Supervised-learning (SL) neural network to tackle these problems with certain success.



a) Bifurcation chart of Wang-Oscillator



b) Bifurcation chart of Lee-oscillator

Fig. 1. Bifurcation charts of Wang-oscillator vs Lee-oscillator. a) Bifurcation chart of Wang-Oscillator, b) Bifurcation chart of Lee-oscillator.

However, when it comes up to massive input data such as financial prediction with over thousands time series financial data and signals as input vectors, these conventional ANNs using classical sigmoid-based activation function are usually “trapped” in local maxima (or minima) during neural network training; which not only affect the efficiency (time cost), but also the accuracy of the forecast results.[18]

3.2. COSMOS CSL network − chaotic FFBP-based SL network for time-series financial prediction

In short, COSMOS Chaotic Supervised-Learning (CSL) Network is the integration of Lee-oscillator with classical FFBPN by replacing all the neurons with Lee-oscillators. Figs. 2 and 3 show the system architecture and chaotic FFBP-based time-series supervised-learning algorithm of COSMOS CSL Network for financial prediction.

4. Chaotic RBF-based Actor-Critic Reinforcement-Learning networks - optimization of trading strategy

4.1. An overview of Reinforcement-Learning (RL)

Different from supervised-learning model with well-defined “input/target-output” pairs to train the network, there are many situations in which input/target-output pairs do not existed. For example, in stock investment, even-though we have the best stock forecast to tell us when to set the buy/sell bid, but “How much we should invest?” and “When to close the bid?” in order to get the highest returns is a typical optimization problem without exact solution. In that case, we can make use of Reinforcement-Learning (RL) method.

Reinforcement Learning (RL) theory is originated from behavior psychology. Its main concept is to train the neural network with the adoption of feedback signals namely Reinforcement-Signal (RS). For the right behavior, the network will respond with a positive RS to “award” the RL network; while for the wrong behavior, the network will respond with a negative RS to “punish” the RL network. As we can see, RL networks don't need well-defined input/target-output pairs, all they need to do is to search for a set of optimal weights to minimize the negative reinforcement signals.

Classical RL model such as Markov Decision Process (MDP) using stochastic-based reinforcement learning algorithms such as Q-Learning, Dynamic-Programming (DP) and TD-Learning (TDL) with certain success.[14] However, when it come up with time-series financial optimization problems with over thousands of input signals and possible trading strategies, these classical stochastic RL methods are either too computational intensive or difficult to adopt for actual implementation.

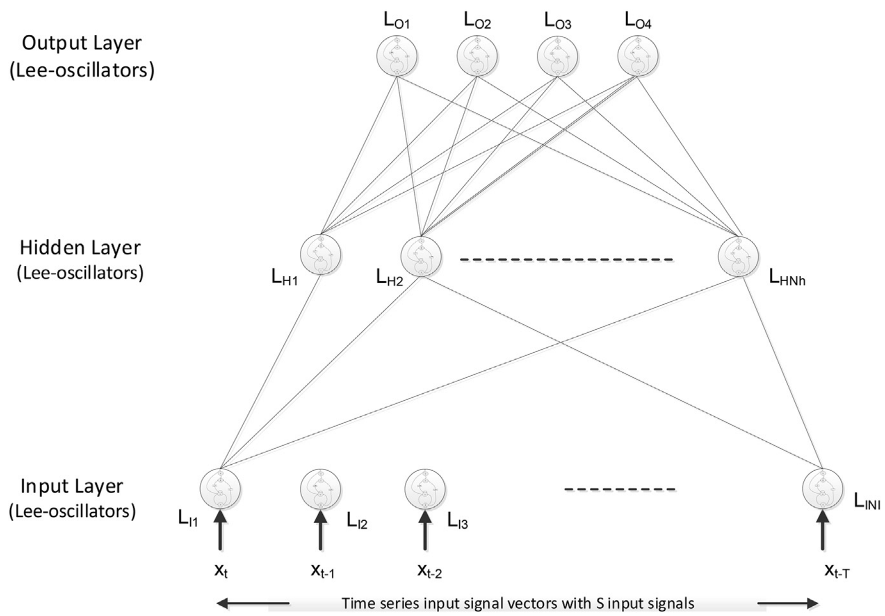


Fig. 2. System architecture of COSMOS CSL network for time series financial prediction.

COSMOS CSL NETWORK LEARNING ALGORITHM

1 COSMOS CSLN Initialization Phase

1.1 Initialization all the network weights ω by a random number generator to values between 1 and 0.

2 COSMOS CSLN Checking Stop Training Criteria

IF MSE < Training Threshold δ (say $1 x 10^{-6}$) STOP,
Else CONTINUE

3 COSMOS CSLN Forward Propagation Phase

3.1 Evaluate the total inputs for all hidden Lee-oscillators (L_H)

$\overrightarrow{L_{Hinput}} = \sum_{n=0}^{N_I} \overrightarrow{L_{In}}\,\overrightarrow{\omega_n}$

Noted that $N_I = T \times S$ is the total number of input Lee-oscillators, where T is the forecast horizon and S is the dimension of the input signal vector.

3.2 Evaluate the TCAF values of all $\overrightarrow{L_{Hinput}}$ vectors using chaotic Lee_operator given by equations (5) to (8)

$\overrightarrow{L_H} = \overrightarrow{Lee_H(L_{Hinput})}$

3.3 Evaluate the total input vectors all output Lee-oscillators (L_O)

$\overrightarrow{L_{Oinput}} = \sum_{n}^{N_H} \overrightarrow{L_{Hn}}\,\overrightarrow{\omega_n}$

Noted that N_H is the total number of hidden Lee-oscillators.

3.4 Evaluate the TCAF values of all $\overrightarrow{L_{Oinput}}$ vectors

$\overrightarrow{L_O} = \overrightarrow{Lee_O(L_{Oinput})}$

4 COSMOS CSLN Backward Propagation Phase

4.1 Evaluate the $\vec{\delta}_o$ (Correction Error Vector) and $\overrightarrow{\Delta\omega_{HO}}$ (weight adjustment vectors between hidden and output layer) of all $\vec{L}_o$ against the target output vectors $\vec{L}'_O$ with network learning rate β.

$\overrightarrow{\delta_{HO}} = (\vec{L}'_O - \vec{L}_O) f'_{L_O}(\overrightarrow{L_{Oinput}})$

$\overrightarrow{\Delta\omega_{HO}} = \beta\overrightarrow{\delta_{HO}}\,\overrightarrow{L_H}$

4.2 Evaluate the $\vec{\delta}_H$ and $\overrightarrow{\Delta\omega_{IH}}$ of all $\vec{L}_H$

$\overrightarrow{\delta_{IH}} = [\sum \overrightarrow{\delta_{HO}}\,.\,\overrightarrow{\omega_{HO}}](\vec{L}'_O - \vec{L}_O) f'_{L_O}(\overrightarrow{L_{Oinput}})$

$\overrightarrow{\Delta\omega_{IH}} = \beta\overrightarrow{\delta_{IH}}\,\overrightarrow{L_I}$

4.3 Evaluate the all the weight vectors at the same time.

$\overrightarrow{\omega(t+1)} = \overrightarrow{\omega(t)} + \overrightarrow{\Delta\omega(t)}$

5 COSMOS CSLN STEP 2 to Check for Stopping Criteria.

Note:
1. L_I, L_H and L_O are the Lee-oscillator in the input, hidden and output layer.
2. The four output Lee oscillators correspond to the next-day forecasts of Open, High, Low and Close.
3. ω are the network weights.
4. TCAF – Transient Chaotic Activation Function.
5. δ are the correction error vectors.

Fig. 3. COSMOS CSL network learning algorithm.

With the flourishing of recurrent neural networks in the past decades, researchers start to explore how recurrent neural networks can be applied for RL on various optimization problems.[20,22]

4.2. Discrete-Time Actor-Critic RL model

A typical time-series RL model can be visualized as a Discrete-Time Actor-Critic RL Model (DTAC-RLM) (Fig. 4).

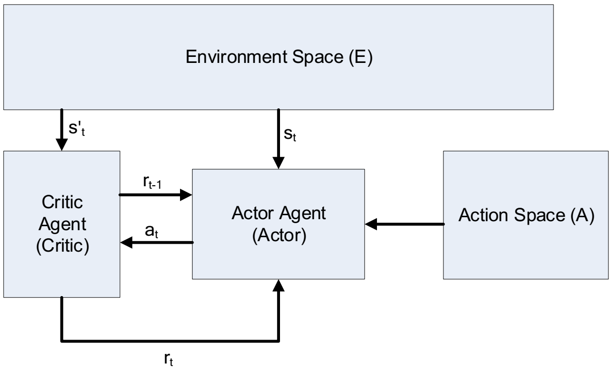


Fig. 4. Discrete-time actor-critic RL model.

DTAC-RLM is a multi-agent system[18] consists of four components: Environment Space (E), Action Space (A), Actor Agent (Actor) and Critic Agent (Critic). Different from its Continuous-Time AC model counterpart (such as MDP), DTAC-RLM visualizes its "world" as a collection of discrete-time-step states and actions. More vitally, all these states are related as time-series events. The role of the Actor is based on the input signals provided at current state at time t (s_t) to response with the "best" available actions (a_t); whereas the role of the Critic is to evaluate the Reinforcement Signal (RS, or "reward, r_t" in short) based on the action(s) taken by the Actor and the signals provided by the new states (s_{t+1}). Then the reward signal r_t feedbacks as input to the Actor to decide the next action **a** at time t+1.

The formulations are given by:

$$\boldsymbol{s} \in \{\boldsymbol{S}\}, \boldsymbol{a} \in \{\boldsymbol{A}\}, \boldsymbol{r} \in \{\mathbb{R}\} \tag{9}$$

$$\boldsymbol{\Pi} = \{\boldsymbol{a_t}, \boldsymbol{t} = 1..\boldsymbol{T}\} \tag{10}$$

$$V = \sum_{t=1}^{T} r_t(s_t, a_t) \tag{11}$$

$$\mathbf{V}^* = \max_{\boldsymbol{\Pi}} \boldsymbol{V}^{\boldsymbol{\Pi}} \tag{12}$$

As shown in the above formulation: **S** and **A** denote the State Space and Action Space; **s** and **a** denote the state and action vectors, **r** denotes the reward vector; Π denotes the action policy from time t = 1 to t = T (Dimension of Policy Space); V denotes the Value Function which represents the total returns for a particular policy Π. For DRL with maxI iterations, there will be totally maxI policies Π being generated, and V* is the optimal returns obtained by the calculation of all possible return values $\boldsymbol{V_k^{\Pi}}$ (where k = 1 … maxI) from these policies. So the whole optimization problem is to find the best policy Π* via DRL in order to attain the optimal returns V*. Next section we will explore how to adopt CRBFN (Chaotic Radial Basis Function Neural Network) for DRL.

4.3. Radial Basis Function Neural Network (RBFN) for RL

Like FFBPN, a typical multi-layer RBFN (Radial Basis Function Neural Network) also consists of: Input, Hidden and Output Layers. Although the network architectures between FFBPN and RBFN are highly similar, the network activations in RBF networks are localized RBF functions such as Gaussian functions, resulting in a much faster training rate.[23]

Unlike FFBPN, input vectors in RBFN distribute the values to the hidden layer neurons uniformly, without multiplying them with weights. The neurons in the hidden layer are presented by RBF neurons using radial-basis-function such as Gaussian function as activation function, which is given by:

$$G(r) = \exp\left[-\frac{r}{2\sigma^2}\right] \tag{13}$$

Different from FFBPN which is only tailored for supervised learning, experimental results revealed that RBFNs can basically approximate any functions. As a result, they are usually known as "universal-approximators". In other words, RBFN can be used for both SL and RL by using RBFN to approximate the V-value function.

However, like FFBPNs, when it comes up with handling massive input data and/or highly chaotic nature such as time series financial forecast using over thousands input values and financial signals as network input vectors, RBFNs also encounter "over-training" and "deadlock" problems which hinder the further improvement of network accuracy and resulted in an expensive time costs from the computational perspective.

4.4. COSMOS CRL network - chaotic RBF-based Actor-Critic RL network for the optimization of trading strategy

4.4.1. System overview

COSMOS Chaotic Reinforcement-Learning (CRL) Network integrates the Lee-oscillator technology with conventional RBFN by: 1) Replacing all the Gaussian-based radial basis functions in the hidden layer of RBFN with Lee-oscillatory RBF functions (namely "LRbf"); and 2) Replacing the output neurons with Lee-oscillator to facilitate transient-chaotic neural activations. Fig. 5 shows the LRbf() which served as the chaotic RBF activation function in COSMOS.

The formulation of LRbf is given by:

$$LRbf(x) = \begin{cases} L(x), & x < c \\ L(2c - x), & x \geq c \end{cases} \tag{14}$$

where L(x) is the Lee-oscillator function given by equation (8) and c is the center of the LRbf().

In contrast with the Gaussian Function counterpart, the LRbf exhibits a progressive-transient-chaotic property in the two RBF activation regions which provides some sort of "controlled-hysteresis" to sort out the over-training and deadlock problems during RL with massive input vectors.

4.4.2. COSMOS CRL Network System Architecture for the optimization of trading strategy

As depicted in Fig. 6, COSMOS CRL Network is a three-layer chaotic RBF-based neural network with input, hidden and output layers. To accomplish Actor-Critic-based RL, the input layer consists of two classes of input vectors: the reward signal vector **r** and input signal vector **x**. The input signal vector **x** composes of all time-series input signals with T denotes the span of the time-series. Reward signal vector **r** contains four reinforcement feedback signals which are generated by the Critic Agent in the previous time step. These four reward signals $r^{T,B}$, $r^{T,S}$, $r^{P,B}$, $r^{P,S}$ correspond to the four actions $a^{T,B}$, $a^{T,S}$, $a^{P,B}$ and $a^{P,S}$ which emulate de-facto short-term trading strategies:

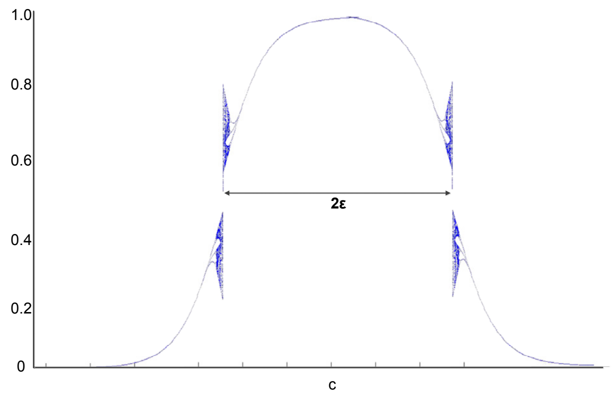


Fig. 5. Chaotic RBF Activation Function using Lee Oscillators (LRbf).

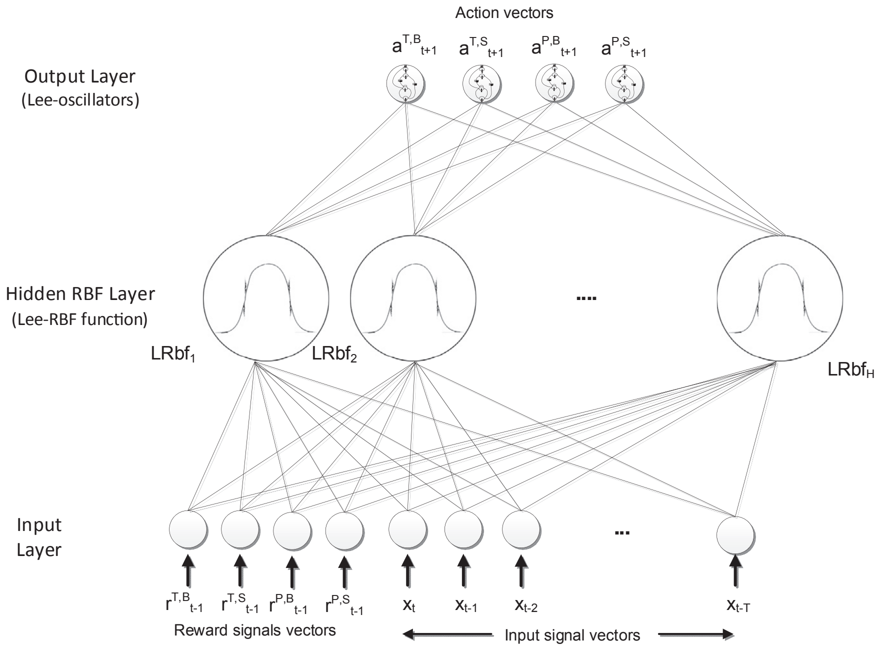


Fig. 6. COSMOS CRL network system architecture.

1) Time-driven-Buy Strategy − Trigger Buy action aT,B when current price reaches forecast Low, day-end harvest;
2) Time-driven-Sell Strategy − Trigger Sell action aT,S when current price reaches forecast High, day-end harvest;
3) Price-driven-Buy Strategy - Trigger Buy action aP,B when current price reaches forecast Low, closing when reaching target price/stop-loss price;
4) Price-driven-Sell Strategy - Trigger Sell action aP,S when current price reaches forecast High, closing when reaching target price/stop-loss price.

The hidden layer of COSMOS CRL Network consists of H LRbf nodes which facilitates transient-chaotic RRB-based RL.

The output layer consists of the four possible actions: $a^{T,B}$, $a^{T,S}$, $a^{P,B}$ and $a^{P,S}$ as described above. Their scalar values are normalized values between 0 and 1 which represent the size (lot) of investment ("0" means "not invest"; "1" means "invest one complete lot").

4.4.3. COSMOS CRL Network Learning Algorithm

The overall COSMOS CRL Network learning algorithm is presented in Fig. 7.

As shown in Fig. 7, the Lee-oscillators are adopted into the algorithm in two aspects: 1) Replacement of the Gaussian RBF with the chaotic RBF (LRbf) in the Hidden Layer; 2) Replacement of Sigmoid-based activation function with Lee-transient-chaotic activation function L in the Output Layer for the determination of the action **a**. From the neural dynamic perspective, the adoption of chaotic RBF and activation function can be considered as the emulation of the chaotic trading behaviors in the time-series events and actions. While from the financial perspective, such adoption of transient-chaotic activation and RBF can be considered as the imitation of "transient-hysteresis" behavior of the investors during real-time trading.

For the Value Function V, the physical meaning of its optimal value V* in terms of financial investment is even more profound: While $\mathbf{r}_t$ is the total returns (rewards) at time-step t which represents the Short-term returns in day-trade; its Value Function V plainly represents the Long-term returns of the whole investment policy Π; and its optimal V* directly represents the optimal Long-term returns by using the optimal COSMOS network configuration and hence the optimal policy Π*.

RACTS - Chaotic Actor-Critic-based DRL Algorithm

1. System Initialization Phase
   1.1 Initialization all the network weights ω between hidden and output layer by a random number generator to values between 1 and 0.
2. For Iteration I = 1 to MaxI, do the following:
3. For each Time-Step t=1 to M, do the following:
4. Input Data Clustering
   4.1 For each hidden RBF oscillator $LRbf_1$ .. $LRbf_H$, cluster all the input vectors x;
   4.2 Determine the cluster centers (c_1 .. c_H) of these LRbf oscillators by using the k-centers algorithm.
   4.3 Evaluate the LRbf amplitudes (ε) by:
   $$\varepsilon_j = \sqrt{\left[\frac{1}{M}\sum_{k=1}^{M}\left\|x_j - x_k\right\|^2\right]} \quad (15)$$
   Where x_k is the M-closest neighbor of x_j
   4.4 Using the computed ε_j to rescale all the LRbf() functions in the hidden layer accordingly.
5. LRbf evaluation
   5.1 Apply LRbf() function (Eqt. 14) to evaluate the chaotic-transfer-function values ($LRbf_1$ .. $LRbf_H$) for all hidden LRbf oscillators.
6. Evaluation of the four action vector elements using the Lee-chaotic Activation Function L(t) (Eqt. 8)
   $$\mathbf{a} = \mathrm{L}\left(\sum_{y=1}^{H} LRbf_y * \boldsymbol{w}\right) \quad (16)$$
   Where w is the weight vector between hidden and output layer.
7. Evaluation of the four rewards ($r^{T,B}$, $r^{T,S}$, $r^{P,B}$, $r^{P,S}$) by using the Reward Function Evaluation Algorithm listed in section 4.4.3.
8. Calculate the total rewards R:
   $$\mathrm{R} = r^{T,B} + r^{T,S} + r^{P,B} + r^{P,S}$$
9. Next Time-step t
10. Evaluate the Value Function VI at I-th iteration
    $$V_I = \sum_{t=1}^{M} R_t \quad (17)$$
11. Next Iteration I
12. Evaluate the optimal Value Function V*
    $$V^* = \max_{1 \le I \le MaxI} V_I \quad (18)$$
13. Record the optimal RACTS network configuration C*.

Note:
1. M is the size of time series dataset.
2. H is the number of hidden nodes.
3. MaxI = 1000 is the maximum number of iterations.

Fig. 7. COSMOS CRL network learning algorithm.

5. System implementation

5.1. COSMOS – TENA 4-layer architecture

From the implementation perspective, COSMOS adopts a 4-layer system implementation architecture namely "TENA" (Fig. 8).

COSMOS TENA System Architecture consists of:

Technology Layer – supports MT4/MT5 [1] platforms and related programming technologies such as Expert Advisors (EAs), COSMOS agent protocol (CAP).

Encryption Layer – supports critical cryptographical technologies include both encryption and blockchain technologies.

Neural Networks Layer – supports COSMOS Chaotic Supervised-learning (CSL) Networks, COSMOS Chaotic Unsupervised-learning (CUSL) Networks and COSMOS Chaotic Reinforcement-learning (CRL) Networks.

Agents Layer – supports intelligent chaotic agent applications include COSMOS autoassociator, forecaster, trading and critic agents.

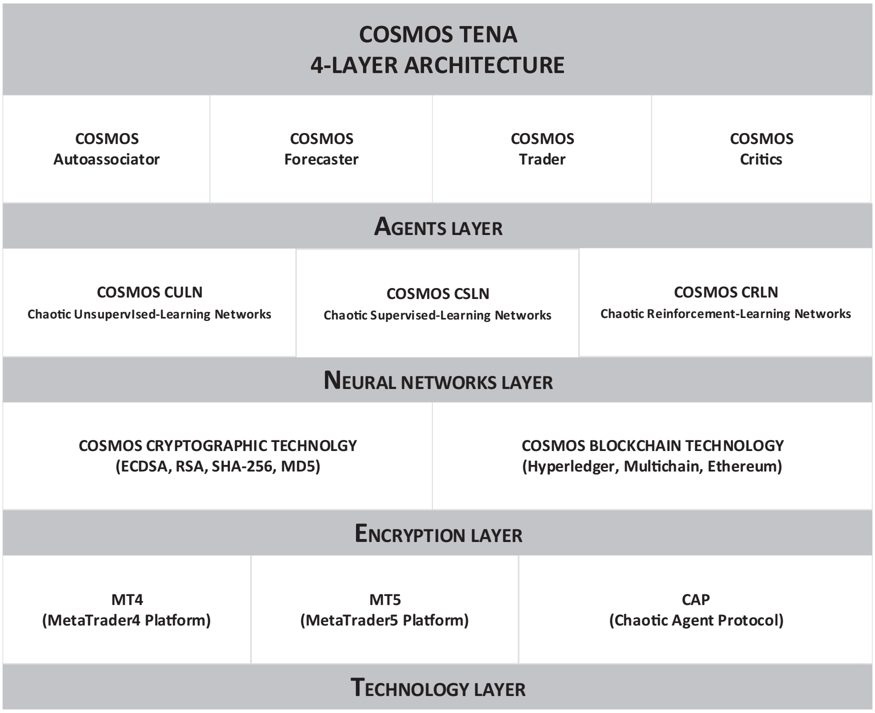


Fig. 8. COSMOS – TENA 4 -Layer System Architecture.

5.2. *COSMOS system implementation using 129 worldwide financial products*

From the application perspective, real-time and historical data of worldwide 129 financial products provided by Forex.com[3] (major online Forex trading platform) and AvaTrade.com[4] (the biggest cryptocurrency trading platform) are adopted for COSMOS system implementation. They include: major cryptocurrencies (9), worldwide forex (84), major commodities (19), worldwide financial indices (17). Table 1 shows the list of financial products under these four categories. As shown in Table 1, owing to the short trading history of cryptocurrencies, 300 trading day records are provided by AvaTrader.com, while all other financial products consist of 2048 past trading day records for each product with data provided by Forex.com which provide sufficient training and test sets for system testing.

5.3. *COSMOS - System Implementation Framework*

For the ease of fully integration and automation of the COSMOS system with both real time and historical financial data provided by Forex.com and AvaTrade.com, the whole intelligent agent-based system fully integrated with MT platforms. Fig. 9 shows the System Implementation Framework of COSMOS.

The whole COSMOS system consists of two main modules: 1) COSMOS Forecaster using COSMOS CSL Network for time-series worldwide financial prediction; and 2) COSMOS Trader and Critic Agents using COSMOS CRL Network for the optimization of trading strategy.

The agent activities of these COSMOS agents are described as follows:

5.3.1. *COSMOS forecaster*

COSMOS Forecaster is a server-side forecast agent located at the server farm of Quantum Finance Forecast Center[2] using Intel i5 CPU 2.39 GHz 32 MB RAM Dell Servers.

For each financial product, 2048-trading day data (except cryptocurrency which only have 300-trading day data) include: Open (O), High (H), Low (L), Close (C) and Volume (V) are automatically generated by MT4 engines of

Table 1
List of 129 financial products.

Product Code	Product Description
Cryptocurrencies (Data provided by AvaTrader.com)	
BCHUSD	BitCoin Cash vs US Dollar
BTCEUR	BitCoin vs Euro
BTCJPY	BitCoin vs JapaneseYen
BTCUSD	BitCoin vs US Dollar
BTGUSD	Bitcoin Gold vs US Dollar
EOSUSD	EOS vs US Dollar
ETH	Ethereum
LTC	Litecoin
XRP	XRP
Financial Index (Data provided by Forex.com)	
AUS200	AUSSIE 200
CHINAA50	China A50 Index
ESP35	Spain 35 Index
ESTX50	EURO STOXX 50 Index
FRA40	CAC 40 Index
GER30	DAX 30 Index
HK50	Hang Seng Index
IT40	Italy 40 Index
JPN225	Nikkei Index
N25	Netherlands 25 Index
NAS100	Nasdaq Index
SIGI	Singapore Index
SPX500	SP500 Index
SWISS20	Switzerland 20 Index
UK100	FTSE 100 Index
US2000	US Small Cap 2000
US30	Dow Jones Index
Commodity (Data provided by Forex.com)	
COPPER	Copper
CORN	Corn
COTTON	Cotton
HTG_OIL	HTG Oil
PALLAD	Palladium
PLAT	Platinum
SOYBEAN	Soybean
SUGAR	Sugar
UK_OIL	Brent Crude Oil
US_NATG	US Natural Gas
US_OIL	WTI Crude Oil
WHEAT	Wheat
XAGUSD	Silver vs US Dollar
XAUAUD	Gold vs Australian Dollar
XAUCHF	Gold vs Swiss Franc
XAUEUR	Gold vs Euro
XAUGBP	Gold vs British Pound
XAUJPY	Gold vs Japanese Yen
XAUUSD	Gold vs US Dollar
Forex (Data provided by Forex.com)	
AUDCAD	Australian Dollar vs Canadian Dollar
AUDCHF	Australian Dollar vs Swiss Franc
AUDCNH	Australian Dollar vs Chinese Yuan
AUDJPY	Australian Dollar vs Japanese Yen
AUDNOK	Australian Dollar vs Norwegian Krone
AUDNZD	Australian Dollar vs New Zealand Dollar
AUDPLN	Australian Dollar vs Polish Zloty
AUDSGD	Australian Dollar vs Singapore Dollar

(*continued on next page*)

Table 1 (*continued*)

Product Code	Product Description
AUDUSD	Australian Dollar vs US Dollar
CADCHF	Canadian Dollar vs Swiss Franc
CADJPY	Canadian Dollar vs Japanese Yen
CADNOK	Canadian Dollar vs Norwegian Krone
CADPLN	Canadian Dollar vs Polish Zloty
CHFHUF	Swiss Franc vs Hungarian Forint
CHFJPY	Swiss Franc vs Japanese Yen
CHFNOK	Swiss Franc vs Norwegian Krone
CHFPLN	Swiss Franc vs Polish Zloty
CNHJPY	Chinese Yuan vs Japanese Yen
EURAUD	EUR vs Australian Dollar
EURCAD	EUR vs Canadian Dollar
EURCHF	EUR vs Swiss Franc
EURCNH	Euro vs Chinese Yuan
EURCZK	EUR vs Czech Koruna
EURDKK	EUR vs Danish Krone
EURGBP	EUR vs British Pound
EURHKD	Euro vs Hong Kong Dollar
EURHUF	EUR vs Hungarian Forint
EURJPY	EUR vs Japanese Yen
EURMXN	Euro vs Mexican Peso
EURNOK	EUR vs Norwegian Krone
EURNZD	EUR vs New Zealand Dollar
EURPLN	EUR vs Polish Zloty
EURRON	Euro vs Romanian Leu
EURRUB	Euro vs Russian Ruble
EURSEK	EUR vs Swedish Krona
EURSGD	Euro vs Singapore Dollar
EURTRY	EUR vs Turkish Lira
EURUSD	EUR vs US Dollar
EURZAR	Euro vs South African Rand
GBPAUD	British Pound vs Australian Dollar
GBPCAD	British Pound vs Canadian Dollar
GBPCHF	British Pound vs Swiss Franc
GBPDKK	British Pound vs Danish Krone
GBPHKD	British Pound vs Hong Kong Dollar
GBPJPY	British Pound vs Japanese Yen
GBPMXN	British Pound vs Mexican Peso
GBPNOK	British Pound vs Norwegian Krone
GBPNZD	British Pound vs New Zealand Dollar
GBPPLN	British Pound vs Polish Zloty
GBPSEK	British Pound vs Swedish Krona
GBPSGD	British Pound vs Singapore Dollar
GBPUSD	British Pound vs US Dollar
GBPZAR	British Pound vs South African Rand
HKDJPY	Hong Kong Dollar vs Japanese Yen
NOKDKK	Norwegian Krone vs Danish Krone
NOKJPY	Norwegian Krone vs Japanese Yen
NOKSEK	Norwegian Krone vs Swedish Krona
NZDCAD	New Zealand Dollar vs Canadian Dollar
NZDCHF	New Zealand Dollar vs Swiss Franc
NZDJPY	New Zealand Dollar vs Japanese Yen
NZDUSD	New Zealand Dollar vs US Dollar
SGDHKD	Singapore Dollar vs Hong Kong Dollar
SGDJPY	Singapore Dollar vs Japanese Yen
TRYJPY	Turkish Lira vs Japanese Yen
USDCAD	US Dollar vs Canadian Dollar
USDCHF	US Dollar vs Swiss Franc

(*continued on next page*)

Table 1 (*continued*)

Product Code	Product Description
USDCNH	US Dollar vs Chinese Yuan
USDCZK	US Dollar vs Czech Koruna
USDDKK	US Dollar vs Danish Krone
USDHKD	US Dollar vs Hong Kong Dollar
USDHUF	US Dollar vs Hungarian Forint
USDILS	US Dollar vs Israeli Shekel
USDJPY	US Dollar vs Japanese Yen
USDMXN	US Dollar vs Mexican Peso
USDNOK	US Dollar vs Norwegian Krone
USDPLN	US Dollar vs Polish Zloty
USDRON	US Dollar vs Romanian Leu
USDRUB	US Dollar vs Russian Ruble
USDSEK	US Dollar vs Swedish Krona
USDSGD	US Dollar vs Singapore Dollar
USDTHB	US Dollar vs Thai Baht
USDTRY	US Dollar vs Turkish Lira
USDZAR	US Dollar vs South African Rand
ZARJPY	South African Rand vs Japanese Yen

Forex.com and AvaTrader.com. Through the Trading Signal Generator, 39 most common trading signals are generated, together with the 2048-trading day data, they are fed into the forecast system for training and forecast of the next-day OPEN (O), HIGH (H), LOW (L) and CLOSE (C). The predicted forecasts of these 129 financial products are stored at the COSMOS Forecast Database for the COSMOS trading agents to access. Table 2 shows the 39 trading signals generated by Trading Signal Generator.

5.3.2. *COSMOS trader*

The COSMOS Trader receives inputs from three sources: 1) Input signals from the markets (stored at the Input Signal Databanks of the COSMOS Server Farm) of the current and past records; 2) Current forecasts stored at the COSMOS Forecast Database of the COSMOS Server Farm; 3) Reward signals from the previous time-step generated by COSMOS Critic Agent. By using the COSMOS CRL algorithm, it evaluates the four possible actions **a**. Updates its trading policy and keep track the market movements in order to trigger the trading actions.

5.3.3. *COSMOS critic agent*

COSMOS Critic Agent receives inputs from two sources: 1) Input signals from the markets of the current and past records; 2) Latest action **a** taken by the COSMOS Trader. By using the reward evaluation algorithm, COSMOS Critic

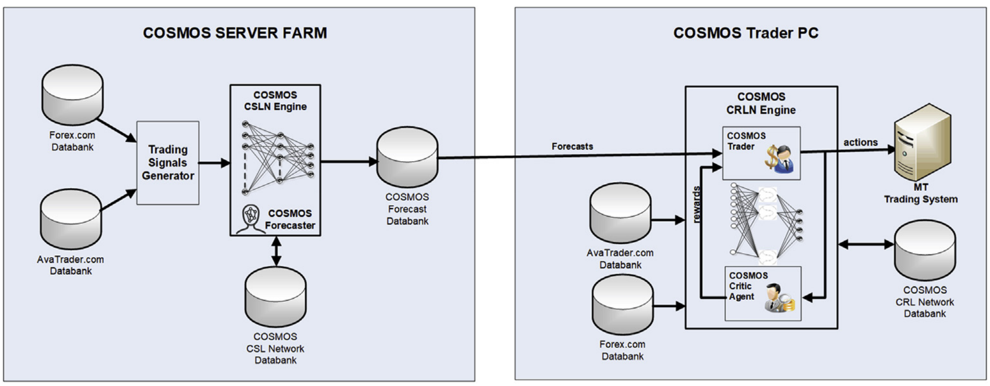


Fig. 9. System implementation framework of COSMOS.

Table 2
List of 39 trading signals.

Code	Signal Name
AC	Accelerator Oscillator
AD	Accu. and Distribution
ADX	Average Directional
ADXWilder	Av Dir by Welles Wilder
Alligator	Alligator
AMA	Adaptive MA
AO	Awesome Oscillator
ATR	Average True Range
BearsPower	Bears Power
Bands	Bollinger Bands
BullsPower	Bulls Power
CCI	Commodity Channel
Chaikin	Chaikin Oscillator
Custom	Custom indicator
DEMA	Double Exponential MA
DeMarker	DeMarker
Envelopes	Envelopes
Force	Force Index
Fractals	Fractals
FrAMA	Fractal Adaptive MA
Gator	Gator Oscillator
Ichimoku	Ichimoku Kinko Hyo
BWMFI	Market Facilitation Index
Momentum	Momentum
MFI	Money Flow Index
MA	Moving Average
OsMA	MACD Histogram
MACD	MACD
OBV	On Balance Volume
SAR	Parabolic Stop & Rev Sys.
RSI	Relative Strength Index
RVI	Relative Vigor Index
StdDev	Standard Deviation
Stochastic	Stochastic Oscillator
TEMA	Triple Exp. MA
TriX	Triple Exp. MA Oscillator
WPR	Williams' Percent Range
VIDyA	Variable Index Dynamic Av.
Volumes	Trading Volumes

Agent evaluates the latest rewards **r** of the four actions and feedbacks to COSMOS Trader as reinforcement signals for the determination of the investment actions in the next time-step.

5.4. COSMOS daily forecast at QFFC

Fig. 10 shows a snapshot of the COMOS Forecaster for the training and forecast of 120 financial products of Forex.com on 09 Nov 2018. As shown in Fig. 10, in a typical daily forecast, the COSMOS Forecaster only takes 62341 msec (62.341 s) to finish the training of forecast of 120 financial products. On the average, it takes 0.519 s (less than 1 s) to complete the network training and forecast process of a single financial product.

Fig. 11 shows the snapshot of COSMOS Forecaster for the system training and forecast of 9 major cryptocurrencies over AvaTrade.com MT platform on the same trading day.

As shown in Fig. 11, COSMOS Forecaster takes 44976 msec (44.976 s) to finish the training and forecast of the 9 cryptocurrencies. On the average, it takes 4.99 s to train and forecast a single cryptocurrency.

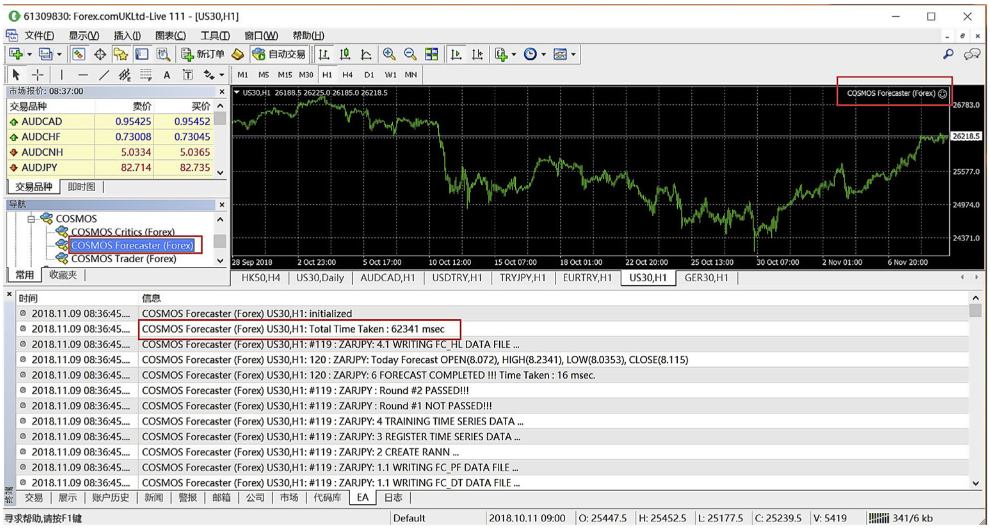


Fig. 10. Snapshot of COSMOS Forecaster for the training and forecast of 120 financial products for Forex.com MT4 platform on 09 NOV 2018.

As compared with all those 120 non-cryptocurrency products, COSMOS Forecaster takes 9.62 times of computer time to predict cryptocurrency, even though cryptocurrency only have 300-trading day records while the other 120 financial products each have 2048-trading day records for system training. It might because cryptocurrencies in general are much more chaotic and fluctuant in nature, which take more time and iterations for COSMOS Forecaster to learn the market patterns.

6. System performance and worldwide evaluation

6.1. COSMOS forecaster performance test

The forecast performance of COSMOS Forecaster is compared with the traditional FFBPN by applying 500 forecast simulations for each system. For the ease of comparison, four categories of worldwide 129 financial products are tested with network MSE (Mean-Square-Error) ranging from 1×10^{-4} to 1×10^{-6} respectively.

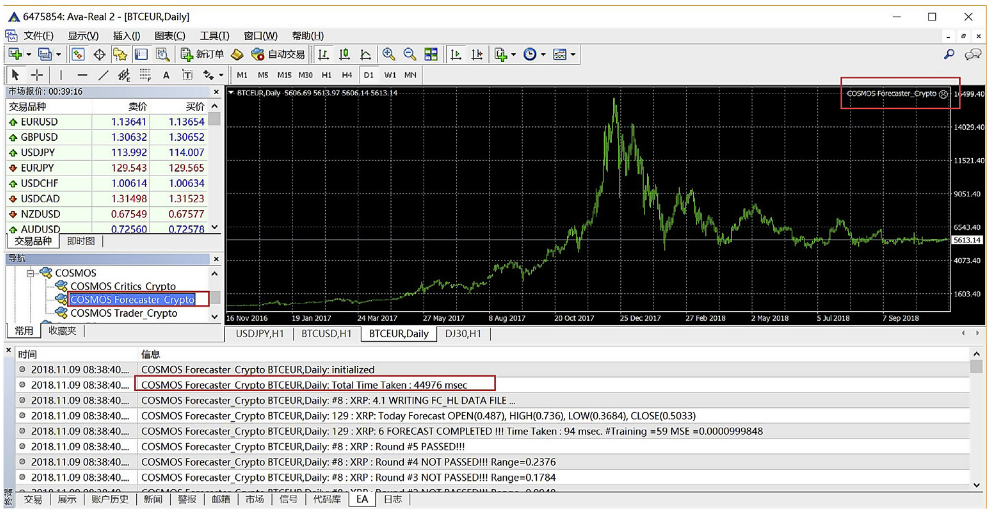


Fig. 11. Snapshot of COSMOS Forecaster for training and forecast of 9 major cryptocurrencies for AvaTrade.com MT4 platform on 09 NOV 2018.

Besides traditional FFPBN, two state-of-the-art financial forecasting methods: i) SVM (Support Vector Machine) forecasting tool provided by R Project (2018), one of the most popular financial forecasting tools used in the finance industry and ii) Deep Neural Network (DNN) with PCA (Principal Component Analysis) model[27] are contrasted with COSMOS Forecaster for forecast performance comparison. Table 3 presents the COSMOS Forecast Performance Comparison Chart of these FOUR systems.

Certain interesting findings are revealed in Table 3:

i) For Case I simulation using MSE 1×10^{-4}, COSMOS Forecaster outperforms FFBPN, SVM and DNN-PCA by 81.29, 53.51 and 40.72 times respectively. Similar findings can be found in Case II simulation results. It clearly reflects the improvement of network learning rate by the adoption of chaotic neural oscillators into traditional FFBPN.

ii) Across the 3 Cases with decreasing MSE from 1×10^{-4}(Case I), 1×10^{-5} (Case II) to 1×10^{-6} (Case III). All forecast systems can achieve the MSE in Case I and Case II. However, for the Case III simulations using MSE 1×10^{-6}, FFBPN encounters "deadlock" problems during the network training of Cryptocurrency and Forex products; while COSMOS Forecaster can still finish the network training with promising training speeds. It clearly demonstrates the resolution of over-training and deadlock problems and sufficient improvement of system training efficiency by COSMOS Forecaster over traditional neural networks using classical sigmoid-based activation function.

iii) In terms of the system performance across different financial products, the simulation results clearly reflect that both cryptocurrency and forex are more chaotic and difficult for network training than other financial products as expected, which will be one of the future R&D directions of Quantum Finance Forecast Center.

6.2. COSMOS trading strategy performance test

In order to conduct a systematic test of COSMOS on intelligent trading, two more benchmark agents are implemented for comparison purposes: 1) Time-driven Trading Agent (TmAgent); 2) Price-driven Trading Agent (PrAgent). These two agents simulate real-world traders by using COSMOS forecasts together with simple short-term

Table 3
COSMOS forecast performance comparison chart.

Product	No. of	FFBPN		SVM		DNN-PCA		COSMOS Forecaster	
Category	PRD	Total STT	Av. STT	Total STT	Av. STT	Total STT	Av. STT	Total STT	Av. STT
Case 1 (MSE = 1×10^{-4})									
Cryptocurrency	9	557251	61916.78	372244	41360.41	244633	27181.47	5720	635.56
Forex	84	508453	6053.01	331511	3946.56	291344	3468.38	6323	75.27
Financial Index	19	41120	2164.21	26152	1376.44	19738	1038.82	923	48.58
Commodity	17	46641	2743.59	29384	1728.46	22108	1300.46	1223	71.94
Overall	129	1153465	8941.59	759291	5885.98	577823	4479.25	14189	109.99
Case 2 (MSE = $1x10^{-5}$)									
Cryptocurrency	9	1460000	162222.22	937320	104146.67	823440	91493.33	15121	1680.11
Forex	84	1235543	14708.85	841405	10016.72	720322	8575.26	17231	205.13
Financial Index	19	111024	5843.37	68391	3599.51	50627	2664.58	2312	121.68
Commodity	17	109142	6420.12	71925	4230.86	44967	2645.09	2640	155.29
Overall	129	2915709	22602.40	1919041	14876.29	1639356	12708.19	37304	289.18
Case 3 (MSE = $1x10^{-6}$)									
Cryptocurrency	9	DL	–	6102255	678028.38	3601307	400145.17	57235	6359.44
Forex	84	DL	–	5477816	65212.10	4184833	49819.44	71243	848.13
Financial Index	19	577324	30385.47	373529	19659.40	318683	16772.78	6623	348.58
Commodity	17	687595	40446.76	468252	27544.25	385741	22690.64	9126	536.82
Overall	129	–	–	12421852	96293.43	8490564	65818.33	144227	1118.04

Note: 1. Results are generated by 500 simulations of each neural network system (measured in msec). 2. "Total STT" denotes the total average system training time for 500 simulations of network training. 3. "Av. STT" denotes the average system training time for a single financial product of each category. 4. "DL" denotes deadlock during system training.

trading strategies of: 1) Day-end harvest strategy (Time-driven Strategy); or 2) Target-profit and Stop-loss strategy (Price-driven Strategy) with ratio 2:1 (i.e. target-profit = 2 × stop-loss). For the 9 cryptocurrency products, 300 trading day simulations are performed for each trading agent; while 500 trading day simulations are performed for the 120 non-cryptocurrency financial products. Table 4 summarizes the trading performance results for three trading strategies.

Major findings can be concluded from Table 4 in three aspects:

i) Across the three trading strategies, overall speaking, COSMOS Trading Strategy outperforms Time-Driven Strategy and Price-Driven Strategy by 106.86% and 88.52% respectively. It clearly reflects the effectiveness of the combined strategy in COSMOS Trader by using chaotic reinforcement-learning technique. Comparing Time-Driven Strategy with Price-Driven Strategy, it is interesting to reveal that Price-Driven Strategy are consistently outperforms its Time-Driven counterpart by around 9.73%. It can be explained by the fact that although Time-Driven Strategy is rather typical in automatic program trading for day-trade, the closing-price normally won't be the optimal price for bid-closing, as compared with the 2:1 Target-profit and Stop-loss Price-Driven Strategy which is more sensible in terms of short-term trading.
ii) Across the four different categories of financial products, the overall returns of cryptocurrency products are the worst as expected, since the forecast performance of cryptocurrency products are also the lowest as compared with other three categories. However, for Forex products, it is interesting to find out that although the forecast performance of forex products is not as good as financial indices and commodities, their overall returns outperforms these two categories and rank no. 1. As reflected by experienced traders, it might be owing to the fact that although the daily patterns of the forex products are highly chaotic, their price patterns are mostly (over 80% of time) oscillations between the predicted HIGH/LOW without trigger the stop-loss thresholds.
iii) As compared with the BUY vs. SELL strategy, it is interesting to find out that the overall returns from SELL strategy are consistently higher that its BUY strategy counterpart by around 16%. Which is quite consistent with the comments from experienced traders that automatic program trading usually favors SELL vs. BUY strategy.

6.3. COSMOS worldwide trading strategy evaluation

Quantum Finance Forecast Center[2] is a non-profit making, self-funded AI-Fintech R&D and worldwide financial forecast center aims at the R&D and provision of a fair and open platform for worldwide traders and individual investors to acquire free knowledge of worldwide financial forecasts based on state-of-art AI technologies. QFFC launched the 129 financial products daily and weekly forecast services from 1 Jan 2018 for over 1000 worldwide traders and individual investors for testing and evaluations. Fig. 12 shows the official site of Quantum Finance Forecast Center.

6.3.1. Past 500-day forecast performance

From forecast performance and evaluation perspectives, COSMOS system evaluates the daily forecast performance of the 129 financial products in four timeframes: daily, weekly average, monthly average and past 500-day average. Fig. 13 shows the past 500-day COSMOS forecast performance ranking list of the top 20 financial

Table 4
COSMOS trader trading strategy performance comparison chart.

Product	Time-Driven Strategy			Price-Driven Strategy			COSMOS Trader Strategy		
Category	BUY	SELL	Overall	BUY	SELL	Overall	BUY	SELL	Overall
Cryptocurrency	2.66%	3.17%	2.92%	3.12%	3.82%	3.47%	6.71%	9.37%	8.04%
Forex	5.52%	7.22%	6.37%	6.32%	7.76%	7.04%	12.41%	13.83%	13.12%
Financial Index	5.72%	6.11%	5.92%	5.49%	6.71%	6.10%	9.56%	10.21%	9.89%
Commodity	4.39%	6.01%	5.20%	5.12%	6.43%	5.78%	10.43%	11.88%	11.16%
Average	4.57%	5.63%	5.10%	5.01%	6.18%	5.60%	9.78%	11.32%	10.55%

Note: 1. Results (Monthly Returns %) are generated by 300 trading day simulations of the 9 Cryptocurrencies and 500 trading day simulations of the 120 non-Cryptocurrency products.

Fig. 12. The official site of Quantum Finance Forecast Center with COSMOS Daily Forecast on 09 Nov 2018.

products. As shown in Fig. 13, the 500-day average forecast % error of the top 20 financial products are ranging from 0.014% to 0.227% respectively, which is rather promising and significant as reflected by over 1000 + members of QFFC which consists of professional forex traders, quants and investors.

6.3.2. *Worldwide trading strategy performance evaluation*

Started from 1 Jan 2018, 342 members of QFFC, which consists of professional traders and quants from major fund houses are invited to join the COSMOS system evaluation focus group for a ONE-YEAR worldwide evaluation program of COSMOS trading system as compared with their own trading strategies. They are all provided with free COSMOS daily forecasts for the 129 financial products and the two de-facto agent trading programs (i.e. the Time-Driven and Price-Driven Trading Agents). Based on the COSMOS daily forecasts, they are free to use either the de-facto trading agents or their own trading strategies to do the trading. Table 5 presents the Nine Months (1.1.2018−30.9.2018) trading performance comparison chart for the worldwide evaluation scheme.

For the ease of comparison, COSMOS Trader trading performance (overall 10.55%) is compared with: 1) Traditional FFBPN with day-trade algorithm (overall 3.07%); 2) SVM forecasting tool with day-trade algorithm (overall 7.53%); 2) DNN-PCA forecasting model with day-trade algorithm (overall 7.88%); 3) Overall average performance of the focus group (overall 4.11%); 4) Top 15% traders' performance (overall 10.47%); 5) Top 5% traders' performance (overall 14.35%).

As revealed from the performance analysis, it is clear that the overall trading performance of COSMOS Trader outperforms FFBPN, SVM and DNN-PCA by around 7.49%, 3.03% and 2.67% respectively (in terms of overall returns). On the other hand, as compared with professional traders, COSMOS Trader trading performance is close to a typical Top 15% trader, outperforms the average traders by over 6.44%, which is rather promising as feedbacks from the professional traders.

Past 500+DAY TOP 3 EURDKK USDHKD EURHUF

Forecast Performance Leaderboards

Number	Product Name	Product Code	HIGH(Error)	LOW(Error)	Average(Error)	%Error
1	EUR/Danish Krone	EURDKK	0.00099	0.00115	0.00107	0.014%
2	US DollarHong Kong Dollar	USDHKD	0.00425	0.00452	0.00438	0.056%
3	EURHungarian Forint	EURHUF	0.40746	0.40363	0.40542	0.129%
4	EURCzech Koruna	EURCZK	0.04256	0.04179	0.04216	0.164%
5	Norwegian Krone/Swedish Krona	NOKSEK	0.00183	0.00187	0.00185	0.177%
6	US Dollar/Chinese Yuan	USDCNH	0.01149	0.01251	0.012	0.184%
7	Euro/Romanian Leu	EURRON	0.00898	0.00856	0.00876	0.189%
8	Australian Dollar/New Zealand Dollar	AUDNZD	0.00216	0.00212	0.00214	0.198%
9	Australian Dollar/Singapore Dollar	AUDSGD	0.00223	0.00209	0.00216	0.208%
10	Euro/Singapore Dollar	EURSGD	0.00337	0.00344	0.0034	0.214%
11	US Dollar/Thai Baht	USDTHB	0.06895	0.07163	0.07017	0.216%
12	Australian Dollar/Norwegian Krone	AUDNOK	0.01474	0.01399	0.01435	0.23%
13	US Dollar/Swiss Franc	USDCHF	0.00214	0.00233	0.00224	0.23%
14	US Dollar/Singapore Dollar	USDSGD	0.0032	0.003	0.0031	0.23%
15	Gold/Japanese Yen	XAUJPY	328.5	338.578	333.41	0.235%
16	Canadian Dollar/Norwegian Krone	CADNOK	0.01507	0.01473	0.0149	0.237%
17	Australian Dollar/Canadian Dollar	AUDCAD	0.00215	0.00259	0.00237	0.239%
18	Australian Dollar/Polish Zloty	AUDPLN	0.007	0.00755	0.00727	0.261%
19	EUR/Polish Zloty	EURPLN	0.01149	0.01082	0.01115	0.264%
20	Singapore Dollar/Hong Kong Dollar	SGDHKD	0.0163	0.01443	0.01535	0.264%

Note:
1. High (Error) = Abs(HighForecast – HighActual)
2. Low (Error) = Abs(LowForecast – LowActual)
3. Average (Error) = Average(High(Error), Low(Error))
4. % Error = Average(Error) / CloseActual

Fig. 13. Past 500-day COSMOS forecast performance chart.

Table 5
COSMOS Trader Trading Performance comparison chart.

Product	FFBPN	SVM	DNN-PCA	ALL Traders	Top 15% Traders	Top 5% Traders	COSMOS
Cryptocurrency	2.03%	5.08%	5.32%	2.76%	6.54%	8.76%	7.62%
Forex	2.72%	7.91%	8.29%	5.32%	12.31%	16.71%	13.16'%
Financial Index	3.65%	7.59%	7.89%	4.24%	11.22%	15.23%	11.37%
Commodity	3.87%	9.52%	10.01%	4.12%	11.81%	16.71%	12.67%
Average	3.07%	7.53%	7.88%	4.11%	10.47%	14.35%	10.55%

7. Conclusion

This paper devises an innovative chaotic oscillatory multi-agent-based neuro-computing system - COSMOS. From the implementation perspective, COSMOS is integrated with Quantum Finance Forecast Center for the provision of worldwide 129 financial product forecasts and intelligent trading systems.

In fact, for a professional trader and investor, a reliable and effective financial forecast system is only the beginning of the story. A good financial investment also needs: 1) good and effective trading and hedging strategies; 2) stable, logical and rational investment psychology.

Current research of QFFC includes: -

1) Quantum Price Level (QPL) research using quantum field theory.
2) Integration of COSMOS with fractal technology for market trends/patterns mining and prediction.
3) R&D on quantum entanglement of Quantum Finance system on severe financial event modeling and prediction.
4) Design and development of intelligent agent-based hedging and trading systems.

Acknowledgment

The author wishes to thank Forex.com and Avatrade.com for the provision of historical and real time 120 + financial product data over their MT4 R&D and trading platforms. The author also wishes to thank Quantum Finance Forecast Center of UIC for the R&D supports and the provision of the channel and platform qffc.org for worldwide system testing and evaluation. This paper and research project is supported by Research Grant #R201901 of Beijing Normal University-Hong Kong Baptist University United International College (UIC).

References

1. MetaQuotes Software Corp. http://www.metatrader4.com; 2019 Accessed 8 Feb 2019.
2. Quantum Finance Forecast Center. http://qffc.org; 2019 Accessed 8 Feb 2019.
3. GAIN Capital Group LLC (Forex.com). http://www.forex.com; 2019 Accessed 8 Feb 2019.
4. AVA Trade EU Ltd (AvaTrade.com). https://www.avatrade.com; 2019 Accessed 8 Feb 2019.
5. Boroden C. *Fibonacci Trading: How to Master the Time and Price Advantage*. UK: McGraw-Hill Education; 2018.
6. Brown C. *Elliot Wave Principle: Elementary Concepts, Wave Patterns, and Practice Exercises*. New York: Bloomberg Press; 2012.
7. Bulkowski TN. *Encyclopedia of Chart Patterns*. UK: Wiley; 2005.
8. Chang PC, Wang D, Zhou C. A novel model by evolving partially connected neural network for stock price trend forecasting. *Expert Syst Appl*. 2012;39(1):611−620.
9. Chen L, Aihara K. Chaotic stimulated annealing by a neural model with transient chaos. *Neural Network*. 1995;8(6):915−930.
10. Dai W, Wu JY, Lu CJ. Combining nonlinear independent component analysis and neural network for the prediction of Asian stock market indexes. *Expert Syst Appl*. 2012;39(4):4444−4452.
11. Dabhi VK, Chaudhary S. Financial time series modeling and prediction using postfix-GP. *Comput Econ*. 2016;47(2):219−253.
12. Henrique BM, Sobreiro VA, Kimura H. Stock price prediction using support vector regression on daily and up to the minute prices. *J Financ Data Sci*. 2018;4(3):183−201.
13. Hwang H, Oh J. Fuzzy models for predicting time series stock price index. *Int J Control Autom Syst*. 2010;8(3):702−706.
14. Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: a survey. *J Artif Intell Res*. 1996;4:237−285.
15. Lee RST. A transient-chaotic auto-associative network (TCAN) based on LEE-oscillators. *IEEE Trans Neural Netw*. 2004;15(5):1228−1243.
16. Lee RST. Advanced paradigms in artificial intelligence from neural oscillators, chaos theory to chaotic neural networks. *Aust : Adv Knowl Int*. 2005:303−380.
17. Lee RST. Lee-associator − a transient chaotic autoassociative network for progressive memory recalling. *Neural Network*. 2006a;19(5):644−666.
18. Lee RST. *Fuzzy-Neuro Approach to Agent Applications (From the AI Perspective to Modern Ontology)*. Germany: Springer-Verlag; 2006b.
19. Lee RST, Liu JNK. Scene analysis using an integrated composite neural oscillatory elastic graph matching model. *Pattern Recogn*. 2002;35:1835−1846.
20. Li CG, Wang M, Huang ZJ, Zhang ZF. An Actor-Critic reinforcement learning algorithm based on adaptive RBF network. In: *IEEE International Conference on Machine Learning and Cybernetics 2009*. 2009:984−988.
21. Ling PW. The stock price forecasting comparative research of the use of fractal theory at Taiwan traditional industry and technology industry. *Appl Mech Mater*. 2013;274:53−56.
22. Liu F, Quek C, Ng GS. Neural network model for time series prediction by reinforcement learning. In: *IEEE International Joint Conference on Neural Networks 2005*. 2. 2005:809−814.

23. Markopoulos AP, Georgiopoulos S, Manolakos DE. On the use of back propagation and radial basis function neural networks in surface roughness prediction. *J Ind Eng Int*. 2016;12(3):389−400.
24. Murphy JJ. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Application*. New York: New York Institute of Finance; 1999.
25. Nahil A, Lyhyaoui A. Short-term stock price forecasting using kernel principal component analysis and support vector machines: the case of Casablanca stock exchange. *Procedia Comput Sci*. 2018;127:161−169.
26. Ouahilal M, Mohajir ME, Chahhou M, Badr Eddine EM. A novel hybrid model based on Hodrick−Prescott filter and support vector regression algorithm for optimizing stock market price prediction. *J Big Data*. 2017;4(1):1−22.
27. Singh R, Srivastava S. Stock prediction using deep learning. *Multimed Tool Appl*. 2017;76(18):18569−18584.
28. Wang X. Period-doublings to chaos in a simple neural network: an analytic proof. *Complex Syst*. 1991;5:425−441.
29. Wong MHY, Lee RST, Liu JNK. Wind shear forecasting by chaotic oscillatory-based neural networks (CONN) with lee-oscillator (retrograde signaling) model. *Conf Microelectron Neural Netw Fuzzy Int Joint Conf Neural Netw*. 2008:2040−2047.