Financial Market Trading System With a Hierarchical Coevolutionary Fuzzy Predictive Model

Haoming Huang, Member, IEEE, Michel Pasquier, and Chai Quek, Member, IEEE

Abstract—Financial market prediction and trading presents a challenging task that attracts great interest from researchers and investors because success may result in substantial rewards. This paper describes the application of a hierarchical coevolutionary fuzzy system called HiCEFS for predicting financial time series. A novel financial trading system using HiCEFS as a predictive model and employing a prudent trading strategy based on the price percentage oscillator (PPO) is proposed. In order to construct an accurate predictive model, a form of generic membership function named Irregular Shaped Membership Function (ISMF) is employed and a hierarchical coevolutionary genetic algorithm (HCGA) is adopted to automatically derive the ISMFs for each input feature in HiCEFS. With the accurate prediction from HiCEFS and the prudent trading strategy, the proposed system outperforms the simple buy-and-hold strategy, the trading system without prediction and the trading system with other predictive models (EFuNN, DENFIS and RSPop) on real-world financial data.

Index Terms—Filter of unnecessary trade, financial trading system, genetic fuzzy system, hierarchical coevolution, irregular shaped membership function (ISMF), percentage price oscillator (PPO), stock trend prediction, technical indicator.

I. INTRODUCTION

Inspired by successful developments in computational intelligence (CI), the application of CI techniques in finance and economics has become a thriving research area [1]. As an important component in CI, evolutionary computation (EC) has been applied to gaming theory, agent-based economic modeling, and financial engineering [2]. The financial market prediction and trading is one of the main application domains of CI techniques in finance and economics. The primary goal is to use CI techniques to forecast the trend of the financial market and subsequently make benefits from taking preemptive actions according to a successful trading model or strategy. However, predicting the financial market is not an easy task because the financial market is a highly complex and dynamic system which involves the actual interactions taken by millions of individual investors and institutions [3].

Many investors are convinced by the predictability of the financial market and try to make profit through exploiting the analysis of the financial data. In general, there are two main categories of analysis: fundamental and technical. The premise of fundamental analysis is that the investors can take advantage of the imbalances that affect the price of that commodity by accurately analyzing the economic factors such as market demand that affect a particular commodity. In contrast, the traders using technical analysis consider the price as the only true barometer of a market and analyze the historical financial data, such as price and volume [4]. However, on the basis of the “efficient market hypothesis” (EMH) [5], many financial theoreticians question the possibility of predicting the financial market using technical analysis. The assumption of the EMH is that the prices on traded assets already reflect all known information and, therefore, are unbiased in the sense that they reflect the collective beliefs of all investors about future prospects. The EMH implies that the price reflects all the relevant information so that it is impossible to consistently outperform the market by using any information that is already available to the public. However, the hypothesis is highly controversial. From statistical and psychological points of view, some works [6]–[8] have attempted to invalidate the EMH and have shown evidence on the predictability of financial market using technical analysis.

Among the CI techniques, neural networks (NNs) [9] are most extensively employed for the financial market prediction [10], [11]. Saad et al. applied the Time-Delay, Recurrent and Probabilistic Neural Networks (TDNN, RNN, and PNN) to analyze the predictive capability of the networks using several highly volatile and consumer stocks [12]. Moody and Saffell used a direct reinforcement learning approach to train a trading system and claimed to produce better trading strategies than systems utilizing Q-Learning for trading based on forecasting the S&P 500 historical data [13]. Chen et al. used PNN to forecast Taiwan stock index movements and presented two PNN-guided investment strategies to translate the predictions to trading signals [14]. Quek et al. proposed a RNN for predicting the price of gold and British Pound-Dollar futures and options [15].

Although NNs yield promising results in financial market prediction, they are mainly considered as black-box models with their knowledge represented by links and weights. Fuzzy systems (FS) [16] are good at explaining their decisions in the form of readable IF-THEN rules. However, pure FS lacks a learning mechanism. These limitations have been a central driving force behind the creation of neuro-fuzzy systems (NFS), which synergize the human-like reasoning style of fuzzy systems with the learning and connectionist structure of NNs. In recent years, there has been an increasing amount of research applying NFSSs in financial engineering [17]. Thammano [18] used a neuro-fuzzy model to predict future values of Thailand’s largest government-owned bank. He claimed the model could predict investment opportunities during the economic crisis when statistical approaches did not yield satisfactory results. Pantazopoulos et al.
[19] proposed neuro-fuzzy methodologies for stock and option prediction and trading strategies. They claim their system is able to make profitable trading of IBM stock and S&P 500 option. Ang and Quek proposed a rough set based neuro-fuzzy approach named RSPOP for use as a stock predictive model [20]. They claimed that the time-delayed price difference forecast approach yields superior predictive capability than the price forecast approach because it alleviates the inability of the network in predicting outside the range of values of a training data set. Cheng et al. [21] adopted a NFS to predict the actions of the stock investors in anticipation of upcoming events that will affect the stock price. Tan et al. [22] proposed a NFS called GCL for predicting the stock price.

In addition to neuro-fuzzy approaches, EC techniques [23] have been applied in hybrid approaches for financial forecasting. Tsang et al. [24] developed the genetic programming (GP)-based decision support system called EDDIE for discovering the interaction between the financial factors and suggesting financial decisions. They pointed out the performance of EDDIE depends on the quality of the selected inputs. Kim and Han used a hybrid genetic NN approach to predict the moving direction of Korea stock price index in the next day as a classification problem [25]. They tried to reduce the dimensionality of the feature space by using a genetic algorithm (GA) to optimize the thresholds for feature discretization. Their system can achieve an accuracy of about 65% but it has not been used for trading in the experiment. Kuo et al. developed a genetic NFS, with the weights and bias initialized by a GA, to provide fundamental analysis with qualitative political, financial, economic factors, etc. [26]. The output of the fundamental analysis was then integrated with the technical indices through a NN to formulate the final trading decisions. Leigh et al. used GA to select a more informative subset of inputs for the NN so as to improve the correlation between the NN estimated price and actual price [27]. The final system is used to identify the bull flag pattern with pattern recognition and to learn the trading rules from price and volume of the NYSE Composite Index. The paper shows that it yielded superior results by the use of combination of GA and NN. For stock market forecasting, Armano et al. proposed a genetic-neural architecture called Neural-XCS (NXCS), which hybridizes the evolutionary learning system XCS with NN by using NNs as the classifiers in XCS [28]. The technical indicators and historical stock prices are fed into the system to predict the future COMIT and S&P 500 index prices. The results show that their system outperforms the buy-and-hold strategy.

Summarizing existing research works, there are three main factors that affect the performance of a trading system; they are namely: the performance of the predictor, the target of prediction, and the trading strategy. In this paper, we propose a novel financial trading system with a trend prediction model empowered by a hierarchical coevolutionary fuzzy system called HiCEFS. Prediction of the future price as can be commonly found in existing works is proven to be very difficult because the price series is highly noisy and chaotic. The prediction of a less chaotic technical indicator i.e., percentage price oscillator (PPO), is thus explored instead in this paper. A prudent trading strategy is devised to generate proper trading signals according to the predicted PPO. The remainder of this paper is organized as follows. In Section II, the hierarchical coevolutionary fuzzy system HiCEFS is introduced. In Section III, the structure of the financial trading system and the trading strategy are described. In Section IV, experiments using real-world stock data are elaborated to illustrate the performance of the novel coevolutionary fuzzy system-based financial trading system. Finally, the conclusion is drawn in Section V.

II. HiCEFS: A HIERARCHICAL COEVOLVING FUZZY APPROACH

A. Overview of HiCEFS

Fuzzy systems are suitable for financial market prediction because they are able to model the dynamics in the chaotic and noisy financial data and provide system knowledge represented by a set of readable rules. GAs can be applied in fuzzy system construction because they are able to efficiently search large spaces without the need of derivative information and they are less prone to local minima problems than gradient learning techniques [29]. Synthesizing the advantages of both techniques, the hybridization of fuzzy system and GA has yielded encouraging results [30–32].

The HiCEFS predictive model is an evolutionary fuzzy system, illustrated in Fig. 1. In the HiCEFS, the hierarchical coevolutionary genetic algorithm (HCGA) [33] automatically derives the important components in the fuzzy system with the goals of achieving a fuzzy system with high performance and minimizing the human efforts in system construction process. The input layer in HiCEFS consists of \( I_e \) nodes, which represent the input variables and receive the input instances. The \( \Pi_{i,j} \) nodes in the condition layer contains the linguistic terms for rule premises represented by the ISMFs, which convert the crisp input values into fuzzy membership values for use in subsequent inference steps. The rule layer forms the fuzzy rule base of the HiCEFS. The Mamdani [34] type rules are adopted in HiCEFS because they are more interpretable in contrast to the TSK [35] type. Each \( R_{ji} \) node in the rule layer represents one rule by linking the condition layer nodes as the IF part to the consequence layer node as the THEN part of the rule. The consequence layer consists of the linguistic terms for consequents represented as \( \Omega_{d,k} \) nodes. Lastly, the \( Q_k \) nodes in the output layer give the final system outputs. With the neuro-fuzzy-like architecture, the HiCEFS is able to easily adopt neuro-fuzzy rule generation methods [36] while functioning as a Mamdani-type fuzzy system. As a financial market predictive model, HiCEFS uses the historical financial time series as inputs to predict the future values of the same.

B. Irregular Shaped Membership Function

The proper formation of the fuzzy partitions represented by a set of membership functions (MFs) is a critical step towards a system with satisfactory performance. The fuzzy partitions should accurately and concisely describe the data distribution on the feature domains. This generally entails two requirements: First, the MFs used in the fuzzy partitions should be sufficiently flexible to accurately represent various data distributions. Second, the fuzzy partitions should be concisely formed by a proper number of MFs, because an excessive number of MFs generally results in an overly complex and
noninterpretable system. These two requirements are believed to be highly correlated because the same data distribution can be represented by fewer flexible MFs than by those less flexible ones. For practical reasons, however, most fuzzy systems use a specific type of MF predefined by the designers. It is preferable to employ one generic kind of MFs that can be generated according to the data distribution with minimal human intervention, especially since the latter usually requires lots of design effort and yet results in suboptimal performance. Being able to fulfill these requirements, a generic form of membership function called Irregular Shaped Membership Function (ISMF) [37] is adopted in the HiCEFS predictive model.

The shape and position of the ISMFs are determined by a set of unevenly spaced sampling points. The most important point that roughly determines the position of the ISMF is termed as a pivot point. Each ISMF contains exactly one pivot point, at which the membership value is ensured to reach 1. Other points in an ISMF are positioned on either side of the pivot point and are called shoulder points. The membership value at a shoulder point is less than or equal to 1. The allocation of these sampling points is a very important factor that affects the resultant ISMFs. The property of unevenly spaced allocation of the sampling points ensures that the ISMFs are able to better represent various types of data distribution. The ISMFs can approximate the existing commonly used types of membership functions, such as bell, Gaussian, triangular, or trapezoidal. Furthermore, ISMFs are more flexible than common types because they can contain more sampling points. Therefore, they are able to form shapes that can not be properly represented by common type MFs. Fig. 2 illustrates the possible shapes of the ISMFs. With these main properties, the ISMF provides a generic form of MF that is more capable of representing different types of data distribution. Although flexibility is the main property of ISMFs, decoding schemes are necessary and are carefully devised to ensure that the extracted ISMFs are kept valid with respect to their use within a fuzzy rule inference system.

For an ISMF, the maximum number of shoulder points per side is an important parameter and denoted as $S$. If $S$ is too small, the data representation performance of the ISMF will be significantly degraded. However, if $S$ is too large, the length of the encoded chromosome for the ISMFs will increase significantly and, hence, the search space will be enlarged and the efficiency of evolutionary tuning will be undermined. For the leftmost and rightmost ISMFs of a linguistic variable, each ISMF requires a pivot and at most $S$ shoulder points, giving a total of $2S$ points. For the other ISMFs, each requires a pivot and at most $S$ shoulder points on each side, giving a total of $2&S$ points. The ISMF generation process using HCGA is discussed in the following subsection.

C. Hierarchical Coevolutionary Genetic Algorithm (GA) for Training the HiCEFS

1) Background and Overview: The conventional evolutionary techniques normally evolve a single population of potential solutions. The drawback of these approaches is that
Step 1: Determine the number of population levels $PL$ according to the types of the components that are to be learned by HCGA. Define the number of populations $NP_\alpha$, $\alpha \in \{1, 2, ..., PL\}$ according the number of components. Define the size of populations $SP_\alpha$, the crossover rate $P_\alpha^c$ and mutation rate $P_\alpha^m$, $\alpha \in \{1, 2, ..., PL\}$ from different levels and the number of maximum generation $G_{\text{max}}$.
Initialize all the populations and the generation $G=1$.
Step 2: Initialize the index of the chromosome $c=1$.
Step 3: Select the $c^{th}$ chromosome in the highest level (i.e. Level-$PL$) population and determine the set of lower-level (i.e. Level-1, Level-2, ..., Level-$(PL-1)$) chromosomes selected by it.
Step 4: Decode the possible solution from the set of chromosomes determined in Step 3.
Step 5: Evaluate the decoded solution and assign fitness values to the set of chromosomes determined in Step 3.
Step 6: If $c=SP_{PL}$ ($SP_{PL}$ is the size of Level-$PL$ population), assign proper fitness value to the chromosomes that are not evaluated in current generation, evolve all the populations through crossover and mutation operations, $G=G+1$; else $c=c+1$, go to step 3.
Step 7: If $G=G_{\text{max}}$ or the learning goal achieved, stop the coevolution process and select the best solution; else go to Step 2.

Fig. 3. The evolutionary process of the HCGA.

sometimes they manifest convergence towards local optima and also they involve high computational cost. The problem becomes more obvious when one deals with complex tasks which have strong interdependencies among the components of the solution. Coevolution has served as inspiration to derive a family of coevolutionary algorithms [38], [39], which are capable of surmounting these limitations and widening the range of applications suitable for EC. Similar to their counterparts in biology, coevolutionary algorithms involve a number of independently evolving species which together form well-suited solutions to a problem. The fitness of an individual depends on its ability to collaborate with individuals from other species. These coevolutionary algorithms are highly suitable for handling tasks with increasing requirements of complexity and modularity, while at the same time keeping computational cost bounded.

In the HiCEFS predictive model, the hierarchical coevolutionary genetic algorithm (HCGA) [33] serves as a major learning mechanism. The most significant difference between a conventional GA and the HCGA is the special structure of the latter. Most existing GAs encode the whole problem solution into a single chromosome, which can be very long. The HCGA in HiCEFS adopted a novel approach of partitioning and encoding the possible solutions in populations at different levels, allowing for different kinds of chromosomes and genetic operators. A higher level chromosome will select a set of lower-level chromosomes to form a possible solution. In this case, a highly complicated search task can be properly partitioned into several subtasks, which are simultaneously and effectively handled by HCGA. Populations from different levels in HCGA are used to evolve different components in the HiCEFS. The HCGA in HiCEFS is implemented as a two-level hierarchical module, illustrated in Fig. 4, and is adapted to automatically generate ISMFs for all input features. Assume that the HiCEFS is to be constructed for a problem with $I$ features, $I$ Level-I populations and 1 Level-II population will be coevolved in the HCGA. Each Level-I population is assigned to evolve the fuzzy partition of a feature domain. More precisely, the $i^{th}$ Level-I population is specialized to evolve the ISMFs that form the fuzzy partition on the $i^{th}$ feature domain. The Level-II population is assigned to evolve the fuzzy partition selection table, which searches for the best combination of fuzzy partitions.

Fig. 4 shows the genetic segment of an ISMF and part Fig. 5(b) briefly illustrates the process of determining the shape of the ISMF according to the encoded offsets. In the ISMF genetic segment, the first gene represents $x_{0_{HI}}$, the offset of the pivot from the pivot of the previous ISMF, as shown in Fig. 5(b). If the ISMF is the leftmost one, the first gene $x_{0_{HI}}$ encodes the offset of its pivot from the minimum of the input domain. Following the first gene, there is a total of $JS$ genes for $S$ left and $S$ right shoulder points of the ISMF. Each pair of nearby genes encodes the horizontal and membership value offsets. For instance, the $x_{0_{HI}}$ is the horizontal offset and the $y_{0_{HI}}$ is the membership value offset of the $i^{th}$ left shoulder point $s_{n_{HI}}$ from the $(i-1)^{th}$ left shoulder point $s_{n_{HI}}-1$ in the $i^{th}$ ISMF. The first left (right) shoulder point $s_{n_{HI}}$ ($s_{n_{rI}}$), the $x_{0_{HI}}$ and $y_{0_{HI}}$ ($x_{0_{rI}}$ and $y_{0_{rI}}$) are its horizontal and membership value offsets from the pivot point $p_{HI}$. All these encoded offsets in Level-I chromosomes are the per-unit offsets [37], by which...
the absolute offsets from different range are mapped within the range of [0, 1]. They are adopted here because they make the genetic representation more generalized and the implementation simpler. Fig. 5(b) illustrates how the position and shape of an ISMF are determined according to the encoded offsets, where \( p_n \) is the pivot point, \( s_{n,L,j} \) and \( s_{n,R,j} \) are the left and right shoulder points in the \( n \)th ISMF.

Each chromosome in the Level-II population consists of \( I \) genes, each of which is an integer within the range of \([0, K]\), where \( K \) equals to the size of the Level-I population \( |SP| \). Given \( FP \) as an integer value in the \( j \)th gene in a Level-II chromosome, the \((FP)_j\)th Level-I chromosome (fuzzy partition on one feature domain) in the \( j \)th Level-I population (for the \( j \)th feature) will be chosen to form the fuzzy partition. If the value \( FP_j \) is zero, the \( j \)th feature will be discarded. At Step 3 in Fig. 3, the set of Level-I chromosomes selected by the \( j \)th Level-II chromosome can be determined according to its gene content using the method mentioned above. In this way, the Level-II population selects the most informative features and keeps track of the best combinations of the fuzzy partitions, which are evolved by the Level-I populations, for the final HiCEFS.

After the combination of the Level-I chromosomes is found by decoding the Level-II chromosome, the ISMFs represented by these Level-I chromosomes are to be decoded. Before decoding the ISMFs on a feature domain, the minimum \( d_{\text{min}} \), the maximum \( d_{\text{max}} \), and the range \( R_D \) of the domain are computed with (1), where \( x_{ij} \) is the \( j \)th input variable of the \( j \)th training sample. To decode an ISMF, the position of its pivot point on the feature domain is the first parameter to be decoded. For the pivot of the \( n \)th ISMF, its per-unit offset from the minimum of the domain \( d_{\text{min}} \) is denoted as \( p_{\text{on}} \). It is determined with (2), where \( p_{oi} \) is the encoded offset of the pivot of the \( i \)th ISMF. If there is an ISMF with a pivot point where \( p_{\text{on}} > 1 \), the ISMF will be discarded when constructing the HiCEFS because that means it exceeds the maximum of the range of the domain. If the number of adopted ISMFs is equal to one, the corresponding input feature will be discarded. This property ensures that the HCGA adaptively decides the proper number of membership functions for each feature.

\[
\begin{align*}
    d_{\text{min}} &= \min_j(x_{ij}) \\
    d_{\text{max}} &= \max_j(x_{ij}) \\
    R_D &= d_{\text{max}} - d_{\text{min}} \\
    p_{\text{on}} &= \sum_{i=1}^n p_{oi}
\end{align*}
\]  

(1)

After determining the position of the pivot point, the position of shoulder points can be decoded by mapping the horizontal and vertical per-unit offsets to the \( x \)-axis and membership values. For the left shoulder points of the \( n \)th ISMF, the horizontal per-unit offset is encoded as a proportion of the interval \( R_{PL,n} \) between \( p_{n-1} \) and \( p_n \). For the right shoulder points of the \( n \)th ISMF, the horizontal per-unit offset is encoded as a proportion of the interval \( R_{PR,n} \) between \( p_n \) and \( p_{n+1} \). The \( R_{PL,n} \) and \( R_{PR,n} \) are defined in (3). The per-unit offset of the shoulder points on the range \( R_D \) can be computed using (4), where \( x_{on,t,i} \) is the horizontal per unit offset of the \( t \)th left shoulder point of the \( n \)th ISMF from the range minimum \( d_{\text{min}} \) on \( R_D \); \( x_{on,t,r} \) is the horizontal per unit offset of the \( t \)th right shoulder point of the \( n \)th ISMF from the range minimum \( d_{\text{min}} \) on \( R_D \).

\[
\begin{align*}
    R_{PL,n} &= p_{n} - p_{n+1} \\
    R_{PR,n} &= p_{n+1} \\
    x_{on,t,i} &= p_{\text{on}} - R_{PL,n} \cdot \sum_{i=1}^n x_{on,t,i} \\
    x_{on,t,r} &= p_{\text{on}} + R_{PR,n} \cdot \sum_{i=1}^n x_{on,t,r}
\end{align*}
\]  

(3)

(4)

Based on these offsets, the \( x \) axis value and membership value of each sampling point of an ISMF can be determined using (5)–(7), where \( d_{\text{min}} \) is the minimum value of the feature domain \( R_D \); \( p_{n} \) and \( p_{n+1} \) are the \( x \) axis and membership values, respectively, of the pivot point of the \( n \)th ISMF; \( x_{on,t,i} \) and \( y_{on,t,i} \) are
the \( x \) axis and membership values, respectively, of the \( n \)th left shoulder point of the \( n \)th ISMF; \( x_{n,lu} \) and \( y_{n,lu} \) are the \( x \) axis and membership values, respectively, of the \( n \)th right shoulder point of the \( n \)th ISMF

\[
\begin{align*}
px_n &= d_{\text{min}} + R_D \cdot pd_n \\
py_n &= 1 \\
x_{n,lu} &= d_{\text{min}} + R_D \cdot x_{n,lu} \\
y_{n,lu} &= 1 - \sum_{j=1}^{n} y_{n,j} \\
x_{n,rv} &= d_{\text{min}} + R_D \cdot x_{n,rv} \\
y_{n,rv} &= 1 - \sum_{j=1}^{n} y_{n,j}.
\end{align*}
\]  

(5)  

(6)  

(7)

Similar to the special property of automatically determining proper number of ISMF on a feature domain, the HCGA also attempts to select proper number of sampling points for each individual ISMF. To achieve this special property, a repairing scheme that requires little computational cost is devised. Assume \( x_{n,lu} < px_{n-1} \) or \( y_{n,lu} < 0 \) (\( x_{n,rv} > py_{n+1} \) or \( y_{n,rv} < 0 \)), the left shoulder point \( s_{n,lu} \) (right shoulder point \( s_{n,rv} \)) will be shifted to a proper location by the repairing scheme. The shoulder point \( s_{n,lu} \) having \( i > u \) (\( s_{n,rv} \) having \( j > v \)) will be not be included in the decoded ISMF. More detail on the repairing scheme can be found in [37].

D. Coevolution Process of HCGA

The evolution process of the HCGA shown in Fig. 3 is discussed in greater detail in this subsection. After the HCGA is initialized in steps 1 and 2, the HCGA starts to coevolve the populations at different levels. In Step 3, the \( n \)th Level-II chromosome is selected for fitness evaluation. According to the gene contents of the \( n \)th Level-II chromosome, the set of Level-I chromosomes selected by it can be determined. In Step 4, the ISMFs are decoded from the selected chromosomes. The details of steps 3 and 4 are extensively discussed in Section II-C2. In Step 5, the decoded ISMFs together form the fuzzy partition for the HiCEFS. Owing to the modular structure of the HiCEFS, the fuzzy rule base can be designed manually or derived from data by using rule learning algorithms [36]. In the current version, the rule base is derived from training data using a supervised learning method, namely, the POP rule learning algorithm [40]. (In the next version of HiCEFS, currently under development, the HCGA will directly evolve the rule base as well.) The resultant HiCEFS is then evaluated with the training dataset. The fitness value \( f^S \) for the HiCEFS is computed using (8), where \( R^2 \) is the square of the Pearson product-moment correlation value. The \( f^S \) is assigned to the \( n \)th Level-II chromosome. However, the \( f^S \) cannot be directly assigned to the selected Level-I chromosomes. In a certain generation, a Level-I chromosome can be evaluated for several times because it is selected by more than one Level-II chromosome and participates in more than one HiCEFS. In this case, the fitness assigned to the Level-I chromosome is the maximum fitness obtained by those HiCEFSs it participated in. After evaluating the \( c \)th Level-II chromosome, the coevolution process goes to Step 6 and checks whether all the Level-II chromosomes have been evaluated. If not, the process of steps 3–5 is repeated for evaluating the next Level-II chromosome. Otherwise, the fitness evaluation for the current generation is completed. At this stage, each Level-II chromosome is assigned with a fitness value. However, it is possible that not all Level-I chromosomes are assigned with a fitness value. That is because some Level-I chromosomes may not be selected by any Level-II chromosome in the current generation. For these Level-I chromosomes, the average fitness of the Level-II population is assigned to them so as to give them the proper chance to be selected in future generations. This prevents the HCGA from eliminating good chromosomes and newly generated chromosomes too early. After proper fitness values are assigned to all chromosomes, all populations are evolved into the next generation. Elitism is adopted to preserve the best found fuzzy partition. Assume that the \( c \)th chromosome is the best in the current generation, it will remain intact as the \( c \)th chromosome in the next generation. The roulette wheel selection is used to select the chromosomes for crossover and mutation. The two-point crossover and random mutation is adopted in the Level-II population. A hybrid crossover operator which consists of two crossover cases is tailored for the Level-I populations. The first case is to exchange the corresponding ISMF genetic segments between two parents. In this case, the hybrid crossover attempts to identify superior fuzzy partition without destroying existing ISMFs. The second case is to crossover two parents using (9), where \( a_i, b_i \) are the \( i \)th gene of the two parents, \( a_i', b_i' \) are the \( i \)th gene of the offspring. The \( \sigma^2 \) is set to 0.2 so that the \( \xi_i \) is normally distributed and most likely (with a probability of 97.4%) falls within the range of \([-1, 1]\).

In this case, it achieves balanced exploration and exploitation while keeping all genes in new offsprings within the range of \([0, 1]\). When the hybrid crossover is applied, it randomly picks one from these two crossover cases. The results of preliminary trial runs show that the hybrid crossover only slightly increases the computational time when compared with two-point crossover. A random mutation operator, which replaces a gene with a randomly generated real value in \([0, 1]\), is adopted in the Level-I populations. The evolution process in HCGA terminates when the stopping criterion in Step 7 is met. The final best instance of HiCEFS is used as the predictive model in the financial trading system discussed in the next section

\[
f^S = 100 \times R^2
\]

\[
\begin{align*}
&\begin{cases} 
\alpha_i' = a_i - \xi_i(a_i - b_i) + [b_i - \xi_i(a_i - b_i)] \\
\beta_i' = b_i - \xi_i(b_i - a_i) + [b_i - \xi_i(b_i - a_i)] 
\end{cases}
\end{align*}
\]  

where \( \xi_i \sim N(0, \sigma^2) \).  

(9)

III. FINANCIAL TRADING SYSTEM

A. Technical Analysis for Identifying the Financial Trends

The main approach in financial trading is to identify trends and maintain an investment position (long, short, or hold) until evidence indicates that the trend has reversed. It is evident that trends in stock prices can be very volatile, almost haphazard at times. For identifying the trends of the financial market, investors usually resort to two types of market analysis: fundamental and technical. The primary difference between these two types is that the fundamental analysis is focused on the studies of the causes of market movements, while the technical one is
focused on their effects. The fundamental analysis [41] generally involves the study of the economic forces of supply and demand and often results in long term recommendations. However, three main difficulties are found in fundamental analysis. First, it is difficult to trace back the large amount of relevant historical facts and data to make a complete analysis. Second, the analysis procedures vary from case to case. Thirdly, information must be consistently obtained and the qualitative facts must be correctly interpreted before the rest of the market. The technical analysis [4] assumes that shifts in supply and demand forces will be exhibited in market movements and, therefore, the market movements can be used to determine changes in the ebb and flow of supply and demand. Compared with fundamental analysis, technical analysis can be conveniently conducted for any stock because it only analyzes the historical quantitative data that are easy to obtain, such as the price and volume. Therefore, our financial trading system generates the trading signals according to the results of the technical analysis.

Among various technical indicators, the moving average (MA) is an important one that is used to reveal the trend line by smoothing the cyclical fluctuations in price line and identify trend reversals through crossovers. There are many variations of MAs used in technical analysis. The three most common ones are: simple, weighted and exponential. Compared with the other two MAs, the exponential MA (EMA) assigns the larger weights to more recent periods, which is considered more “adaptive” to the evolving dynamics in stock. A d-day EMA at time t represented by $\text{EMA}_d(t)$ can be derived using (10), where $y(t)$ is the price of the stock at time $t$

$$\text{EMA}_d(t) = \text{EMA}_d(t-1) + \frac{2}{d+1} \left[y(t) - \text{EMA}_d(t-1)\right], \quad (10)$$

Used for signaling the trend reversal, price oscillator is an indicator based on the absolute or percentage difference between a short period MA and a long period MA. The MA and PPO are trend following indicators that can provide trading advice to the investors. In this section, a financial trading system without a predictive model and a financial trading system with HiCEFS predictive model are introduced. In both systems, the trading signal at time $t$ is represented by $F(t)$, where $F(t) \in \{1, -1\}$ with $1$ and $-1$ representing the buy and sell signals, respectively. That means the proposed financial trading system either takes a long or a short position. The price of the stock being traded at time $t$ is denoted as $y(t)$. The transaction cost rate is represented as $\delta$. The multiplicative return $R(t)$ [13], which can be computed using (12), is adopted to measure the final portfolio return of the trading systems. The final multiplicative return $R(T)$ is considered as the important factor to evaluate the performance of the trading systems in this paper

$$R(t) = R(t-1) \left\{1 + r(t)F(t-1)\right\} \left\{1 - \delta |F(t) - F(t-1)|\right\} \quad \text{where} \quad r(t) = \left\{\begin{array}{ll} y(t) & \text{if } y(t) \neq 0 \\ 0 & \text{if } y(t) = 0 \end{array}\right. , \quad (12)$$

The financial trading system without a predictive model is shown in Fig. 6. In this system, the trading signal $F(t)$ is derived according to the current PPO(t), as shown in (13), where $\epsilon$ is the width of the whipsaw signal filter

$$F(t) = \begin{cases} 1, & \text{when } \text{PPO}_{SL}(t) > \epsilon \\ -1, & \text{when } \text{PPO}_{SL}(t) < -\epsilon \\ F(t-1), & \text{else} \end{cases} \quad (13)$$

The MA and PPO are trend following indicators that can identify the current trend but are not able to forecast the trend in the near future. Therefore, the financial trading system without a predictive model always takes the long and short position with a delay of a certain period after the actual trend reversal. In order to undertake prompt trading action by reducing the delay period, the HiCEFS predictive model is adopted in our trading system to forecast the following trend by predicting the future PPO. The overall architecture of the proposed financial trading system with HiCEFS is illustrated in Fig. 7. The PPO series is derived from the historical price series
with (11). Then, the PPO series is represented in $U$-tuples $\{\text{PPO}(t-U+1), \ldots, \text{PPO}(t-1), \text{PPO}(t)\}$, where $U$ denotes the embedding dimension. The $U$-tuples are used as inputs to the HiCEFS predictive model to predict the future PPO at $V$ days later, which is represented as $\text{PPO}^t(t+V)$. In this system, the HiCEFS predictive model is trained using supervised learning, which is discussed in Section II-C, on a set of historical PPO training samples. The trained HiCEFS is then used to predict a set of out-of-sample PPO. All predicted $\text{PPO}^t$ are fed into the trading model which generates trading signals. With the predicted results from HiCEFS, a trading signal is generated based on a modified equation (13) in which the current PPO($t$) is substituted by the predicted PPO($t + V$). This attempts to compensate for the delay in the change in the predicted trend of the trading signal. However, this simple trading rule is overly sensitive because it only considers one prediction output from the HiCEFS. Even though the HiCEFS is able to correctly follow the PPO trend, a single prediction error may trigger a whipsaw in the PPO trend. As a result, the trading system will suffer loss caused by the unnecessary buy/sell trades generated by the trading rule. Therefore, a prudent trading rule, which enables a better risk control by considering a series of latest predicted $\text{PPO}^t$, is devised for our trading system. Considering a trading system that uses HiCEFS to predict the PPO at $V$ days later ($V > 1$), it will go long if all the latest $\lceil V/2 \rceil$ predicted $\text{PPO}^t$ is greater than the filter width $\varepsilon$. Similarly, it will go short if all the latest $\lfloor V/2 \rfloor$ predicted $\text{PPO}^t$ is smaller than $-\varepsilon$. Otherwise, it will just hold the previous trading position. With this improved modification to the trading rule, the accuracy in assessing the predicted trend reversal is greatly improved and the unwanted loss caused by inappropriate trades can be reduced. The computation of the trading signal $F(t)$ is shown in (14), where $\text{PPO}^t_{SL}(t+V)$ is the PPO predicted by HiCEFS, $V$ is the horizon of the prediction and $\varepsilon$ is the width of the whipsaw signal filter $F(t) = \begin{cases} 
abla \varepsilon \in \{V - \lceil V/2 \rceil + 1, \ldots, V\} : \text{PPO}^t_{SL}(t+V) > \varepsilon \\
abla \varepsilon \in \{V - \lfloor V/2 \rfloor + 1, \ldots, V\} : \text{PPO}^t_{SL}(t+V) < -\varepsilon \\
abla F(t-1), \text{ else} \end{cases} \quad (14)$

IV. EXPERIMENTS ON REAL-WORLD FINANCIAL DATA

A. Experimental Setup

In this section, the proposed financial trading system with the HiCEFS predictive model is used to trade the actual index and stock in the real-world financial market. In order to demonstrate the forecasting performance of HiCEFS, three well-known NFSSs including evolving fuzzy neural network (EFuNN) [43], dynamic evolving neural-fuzzy inference system (DENFIS) [44], and rough set-based pseudo outer-product fuzzy neural network (RSPOP) [45], are also used as the predictive model in the experiments. The performance of the proposed trading system with HiCEFS predictive model is benchmarked with the simple buy-and-hold strategy, the trading system without prediction, the trading system with perfect prediction and the trading systems with other predictive models (EFuNN, DENFIS, RSPOP) using the historical data of Hong Kong Hang Seng index (HSI) and Singapore Neptune Orient Lines (NOL) stock.

In the experiments, the width of the whipsaw signal filter $\varepsilon = 0.1\%$ is used in (13) and (14). The proper number of parameter $U$ is estimated by adopting the method of “false nearest neighbors” [46], which is able to determine the embedding dimension of a time series. As shown in Fig. 8, the fraction of false neighbors drastically drops to 0.0056 and 0.0102 at $U = 6$ for HSI and NOL series, respectively. Therefore, $U$ is set to 6 for both HSI and NOL. The parameter $V$ is selected based on two considerations. First, a larger prediction horizon $V$ is
more desirable because the trading strategy can make trading decisions that are more preemptive. However, a bigger challenge will be posed to the predictive models when \( V \) is large. As a result, the prediction accuracy drops, and consequently causes trading losses. Second, we attempt to employ the proposed trading strategy to undertake prudent trading decision based on more than just one prediction output, in which case \( V \) must be greater than 3 according to (14). Therefore, \( V \) is customized to be 5 for the trading system with prediction. That means the predictive models use the PPO values at the most recent six days to forecast the PPO value at five days later. For the trading system without prediction, the trading signals are computed with (11) and (13). For trading systems with a predictive model, the trading signals are generated using (11) and (14).

For trading system with perfect prediction, the trading signal are computed with (11) and (14), but the predicted PPO\((t+i)\) are replaced with the actual future PPO\((t+i)\). According to the price series and the trading signals, the final portfolio value \( R(T) \) can be computed using (12), where the initial portfolio value \( R(0) = 1.0 \) and the transaction cost rate \( \delta = 0.2\% \).

All the predictive models (EFuNN, DENFIS, RSPOP and HiCEFS) are constructed as six-input one-output systems and configured with default parameters. The parameters settings of HCGA in HiCEFS are listed in Table I. The data from historical financial series (PPO series) are partitioned into two nonoverlapping data sets. The first one is the training data set for training the predictive models. The other one is the out-of-sample data set for evaluating the trading systems with the trained predic-

<table>
<thead>
<tr>
<th>HSI</th>
<th>B&amp;H</th>
<th>TS-WOP</th>
<th>TS-EFuNN</th>
<th>TS-DENFIS</th>
<th>TS-RSPOP</th>
<th>TS-HICEFS</th>
<th>TS-PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R(T) )</td>
<td>1.424</td>
<td>1.938</td>
<td>2.106</td>
<td>4.391</td>
<td>2.806</td>
<td>5.781</td>
<td>22.120</td>
</tr>
<tr>
<td>( R^2 )</td>
<td>N.A.</td>
<td>N.A.</td>
<td>0.832</td>
<td>0.905</td>
<td>0.877</td>
<td>0.912</td>
<td>N.A.</td>
</tr>
<tr>
<td>Rules</td>
<td>N.A.</td>
<td>N.A.</td>
<td>354</td>
<td>8</td>
<td>57</td>
<td>16</td>
<td>N.A.</td>
</tr>
</tbody>
</table>
tive models. The EFuNN, DENFIS, and RSPOP are trained with their own learning methods and the trained system is used as the predictive model in the trade systems. The HiCEFS is trained as introduced in Section II-C and adopted as the predictive model in the final trading system. Then, the simple buy-and-hold strategy, the trading system without prediction, the trading system with perfect prediction, and the trading systems with different predictive model (EFuNN, DENFIS, RSPOP and HiCEFS) are evaluated with the out-of-sample data set.

B. Experimental Results and Analysis

1) Hong Kong Hang Seng Index: In this experiment, the trading systems are benchmarked using the Hong Kong Hang Seng index. The experimental data consists of 4941 daily closing price values obtained from the Yahoo finance website on the counter HSI from the period of January 2, 1987 to October 31, 2006. The training data set for predictive models contains the PPO series constructed from 2480 price values before the year 1997. The out-of-sample data set contains the PPO series constructed from the remaining 2461 price values. The PPO series is generated by using (11), with heuristically chosen parameters $S = 15$ and $L = 45$. All the trading systems enter the market with a starting portfolio value of $R(0) = 1.0$.

The out-of-sample price series and the trading signals are shown in Fig. 9. The series of portfolio multiplicative return $R(t)$ for different trading systems are shown in Fig. 10. Table II shows the benchmarking results of different trading systems, including the portfolio end value $R(T)$, square of Pearson correlation $R^2$ between the actual, and predicted PPO series and the number of rules in the predictive model. In Table II, the B&H represents the buy-and-hold strategy; the TS-WOP and TS-PP denotes the trading system without prediction.
TABLE III
THE DERIVED FUZZY RULE BASE IN HiCEFS FOR PREDICTING HSI

<table>
<thead>
<tr>
<th>Rule</th>
<th>PPO(t-5)</th>
<th>PPO(t-4)</th>
<th>PPO(t-3)</th>
<th>PPO(t-2)</th>
<th>PPO(t-1)</th>
<th>PPO(t)</th>
<th>PPO'(t+5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R₁</td>
<td>high</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>slightly high</td>
<td>moderately high</td>
</tr>
<tr>
<td>R₂</td>
<td>high</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>slightly low</td>
<td>slightly high</td>
</tr>
<tr>
<td>R₃</td>
<td>high</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>moderately low</td>
<td>very low</td>
</tr>
<tr>
<td>R₄</td>
<td>high</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>very low</td>
</tr>
<tr>
<td>R₅</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>R₆</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>moderately low</td>
<td>moderately low</td>
</tr>
<tr>
<td>R₇</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>R₈</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>moderately low</td>
<td>moderately low</td>
</tr>
<tr>
<td>R₉</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>moderately low</td>
<td>medium</td>
</tr>
<tr>
<td>R₁₀</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>R₁₁</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>R₁₂</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>R₁₃</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>slightly high</td>
<td>moderately high</td>
</tr>
<tr>
<td>R₁₄</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>slightly high</td>
<td>high</td>
</tr>
<tr>
<td>R₁₅</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>very high</td>
</tr>
<tr>
<td>R₁₆</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
</tr>
</tbody>
</table>

Fig. 13. Price and trading signals on NOL.

and with perfect prediction, respectively; the trading system with EFuNN, DENFIS, RSPOP and HiCEFS are denoted as TS-EFuNN, TS-DENFIS, TS-RSPOP and TS-HiCEFS, respectively. As shown in Table II, the simple buy-and-hold strategy only achieved a final portfolio value \( R(T) = 1,424 \). The trading system without a predictive model yielded a final portfolio value \( R(T) = 1,938 \). The trading system with HiCEFS predictive model managed to achieve a final return of \( R(T) = 5,781 \). Compared with the trading system without prediction, the trading system with HiCEFS achieved an increase of 3.843 in \( R(T) \). Compared with the trading system with EFuNN, DENFIS, and RSPOP, the trading system with HiCEFS yielded an increase of 3.675, 1.390, and 2.975 in final portfolio value \( R(T) \), respectively. Compared with the EFuNN and RSPOP predictive model, the HiCEFS predictive model is able to achieve higher correlation with smaller number of rules. Although DENFIS is able to achieve a similar correlation with only eight rules, the TSK-type rules in DENFIS are less interpretable than the Mamdani-type rules identified by HiCEFS.

The trading signals generated by HiCEFS can be more closely examined with the results in Fig. 11, which is the enlarged part of Fig. 9 from time \( t = 1700 \) to 2010. As shown in part (a) of Fig. 11, the trading system with HiCEFS is able to enter the short selling position earlier than the trading system without prediction because of a successful forecasting case from HiCEFS predictive model. Part (b) of Fig. 11 shows that the trading system with HiCEFS is able to successfully prune some unnecessary trading transactions. It is also possible that the trading system with HiCEFS generates delayed trading signals, as shown in part (c) of Fig. 11. Similar cases can also be found in the trading signals given by the trading systems with other predictive models. This is intuitive because it is impossible to forecast the financial series with perfect accuracy. Therefore, all the trading systems with predictive model achieved lower portfolio end value than the trading system with perfect prediction. The capability of giving more preemptive and less unnecessary or delayed trading signals is the main reason for the higher revenue earned by the trading system with HiCEFS than the trading systems with other
predictive model. The derived membership functions for the inputs and output feature are shown in Fig. 12. Below the MFs, the linguistic terms assigned to the MFs are listed in the order as the MFs appear in the figure. In Table III, the derived fuzzy rules in HiCEFS are also listed.

2) Singapore Neptune Orient Lines: In this experiment, the trading systems are benchmarked using the historical data of the Singapore Neptune Orient Lines (NOL) stock. The experimental data consists of 3960 daily closing price values obtained from the Yahoo finance website on the counter N03.SI from the period of January 2, 1991 to October 31, 2006. The first 1631 price values before July 1997 are used for constructing the PPO series in the training data set. The testing data set contains the PPO series constructed using the 2329 price values from the pe-
period of July 1, 1997 to October 31, 2006. The PPO series is generated using (11), with heuristically chosen parameters of $S = 15$ and $L = 45$. The starting portfolio value $R(0)$ for all trading systems is set as 1.0.

Fig. 13 shows the out-of-sample price series and the trading signals generated by the trading system with the HiCEFS predictive model. Fig. 14 shows the series of portfolio multiplicative returns $R(t)$ for the trading systems without prediction, the trading system with RSPOP and the trading system with HiCEFS. Similar to Table II, Table IV shows the benchmarking results of different trading systems trading on NOL. As shown in Table IV, the simple buy-and-hold strategy only achieved a final portfolio value $R(T) = 1.803$. The trading system without predictive model yielded a final portfolio value $R(T) = 9.153$. The trading system with HiCEFS predictive model managed to achieve a final return of $R(T) = 14.251$. Compared with the trading system without prediction, the trading system with HiCEFS achieved an increase of 5.098 in $R(T)$. Compared with the trading system with EFuNN, DENFIS, and RSPOP, the trading system with HiCEFS yielded an increase of 6.379, 4.019, and 3.745 in final return $R(T)$ against the respective systems. The trading system with EFuNN yielded a $R(T)$ lower than the trading system without prediction. It shows that the predictive model can undermine the performance of the trading system if it fails to correctly predict the trend. As shown in the rows of “$PPO^2$” and “rules” in Table IV, the HiCEFS predictive model is able to follow the stock trend with only 11 Mamdani-type rules and achieve higher correlation than the EFuNN and DENFIS predictive model. We can also find the trading system with RSPOP yielded lower portfolio end value on NOL than the trading system with HiCEFS, even though RSPOP achieved a slightly higher correlation than HiCEFS. This shows that, for improving the profitability of the financial trading systems, the employed predictive model needs to not only follow the overall trends but also give more successful predictions of the trend reversals. The derived membership functions for the input and output features are shown in Fig. 15. Below the MFs, the linguistic terms assigned to the MFs are listed in the order as the MFs appear in the figure. In Table V, the derived fuzzy rules in HiCEFS are listed.

V. CONCLUSION

A novel financial trading system with a predictive model empowered by a hierarchical coevolutionary fuzzy system called HiCEFS is proposed in this paper. A prudent trading strategy using the price percentage oscillator (PPO) as the main technical indicator is adopted to generate trading signals in the proposed system. The main task for the HiCEFS predictive model is to forecast the future PPO in order to take profit by making preemptive trading transactions. To generate an accurate predictive model, the hierarchical coevolutionary genetic algorithm (HCGA) serves as the main learning mechanism for automatically generating the important components in HiCEFS. A generic form of MF termed the Irregular Shaped Membership Function (ISMF) is employed in HiCEFS because it can model more accurately various forms of data distribution. The formation of proper MFs in HiCEFS is divided into the subtasks of generating ISMFs for each input feature which are simultaneously coevolved at different populations in HCGA. The trading system adopting the resultant HiCEFS as the predictive model is benchmarked against trading systems employing other predictive models on real-world stock data. Experimental results confirm that the HiCEFS is able to provide a more accurate prediction of the stock trend and the trading system with HiCEFS is able to yield higher profit than the simple buy-and-hold strategy, the trading system without prediction, and the trading system with other predictive models. However, the authors need to point out that the optimal parameters in the trading strategy including the periods of short and long MA in the computation of PPO usually vary for different stocks. The settings of these parameters can be very crucial to the final profitability of the trading system. Further improvement such as automatically adapting these parameters can be considered as the future research direction. With the next version of HiCEFS evolving the predictive rules as well, it would be possible to tune some trading parameters automatically.

<table>
<thead>
<tr>
<th>Rule</th>
<th>PPO(t-5)</th>
<th>PPO(t-4)</th>
<th>PPO(t-3)</th>
<th>PPO(t-2)</th>
<th>PPO(t-1)</th>
<th>PPO(t)</th>
<th>PPO(t+5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>-</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>$R_2$</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>-</td>
<td>low</td>
<td>slightly low</td>
</tr>
<tr>
<td>$R_3$</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>-</td>
<td>very low</td>
<td>moderately low</td>
</tr>
<tr>
<td>$R_4$</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>-</td>
<td>very low</td>
<td>low</td>
</tr>
<tr>
<td>$R_5$</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>-</td>
<td>high</td>
<td>very high</td>
</tr>
<tr>
<td>$R_6$</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>-</td>
<td>very high</td>
<td>extremely high</td>
</tr>
<tr>
<td>$R_7$</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>-</td>
<td>very high</td>
<td>high</td>
</tr>
<tr>
<td>$R_8$</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>-</td>
<td>high</td>
<td>moderately high</td>
</tr>
<tr>
<td>$R_9$</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>-</td>
<td>medium</td>
<td>moderately high</td>
</tr>
<tr>
<td>$R_{10}$</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>-</td>
<td>medium</td>
<td>slightly high</td>
</tr>
<tr>
<td>$R_{11}$</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>-</td>
<td>medium</td>
<td>slightly high</td>
</tr>
</tbody>
</table>
REFERENCES


Haoming Huang (S’07–M’07) received the B.Eng. degree in computer science and engineering from Zhejiang University, Hangzhou, China, in 2002. He is currently working towards the Ph.D. degree at the Centre for Computational Intelligence, School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include fuzzy systems, evolutionary algorithms, neural networks, decision support systems, and computational intelligence in finance.

Michel Pasquier received a Diploma in electrical engineering and the Ph.D. degree in computer science in 1985 and 1988, respectively, from the National Polytechnic Institute of Grenoble, Grenoble, France. In 1989, he left the LIFIA Laboratory (now INRIA) for the Science City of Tsukuba, Japan, where he was Researcher at the ElectroTechnical Laboratory (ETL) until 1992, and then at Sanyo Electric’s Intelligent Systems Laboratory. In 1994, he joined Nanyang Technological University, Singapore, as a faculty member. Since then, he has been teaching artificial intelligence and other computer science courses, and is the Co-Founder and former Director of the Centre for Computational Intelligence (C2I). He has led funded projects and served as consultant in various application areas, including intelligent robotics and automation, transportation and automotive systems, online learning, as well as financial engineering. His research interests include cognitive architectures and systems, adaptation and learning processes, nature-inspired systems, as well as methods for approximate reasoning, optimization, and planning.