

Integration of mutual information and CRPSO-based fuzzy model for stock index forecasting

Jungwon Yu and Sungshin Kim

Department of Electrical and Computer Engineering
Pusan National University
Busan, Korea
e-mail: {garden0312, sskim}@pusan.ac.kr

Abstract

In this paper, the integration of mutual information (MI) and fuzzy model is proposed to predict stock indexes with complex and non-linear characteristics. Technical indicators are considered as initial input candidates and significant inputs are determined by MI-based input selection method. To identify the structures and parameters of fuzzy models simultaneously, cooperative random learning particle swarm optimization (CRPSO), proposed by Zhao et al., is used. To confirm the effectiveness, the proposed method and comparison methods are applied to the Korea Composite Stock Price Index (KOSPI). The experimental results show that the proposed method, on average, outperforms other comparison methods.

Keywords: Stock index prediction, Mamdani-type fuzzy model, Mutual information, Sequential forward input selection, Cooperative random learning particle swarm optimization (CRPSO)

1. Introduction

Stock index forecasting is one of the most challenging and difficult tasks within the field of financial time-series predictions. Since accurate stock index predictions improve investors' stock market returns, it is a very important research topic and has received great attention. Stock market analysis might be roughly categorized into fundamental analysis and technical analysis. In fundamental analysis, it is assumed that stock index movement depends on corporate profits, macro-economic conditions, political environments, etc. On the other hand, in technical analysis, it is supposed that the required information for stock price prediction has already been reflected in the stock index. Therefore, it is believed that historical index and volume data analysis are all that is needed for stock index forecasting.

Daily stock time-series shows complex, non-linear and non-stationary characteristics because it is influenced by various factors such as economic indicators (e.g. bank rate, bank exchange rate, quarterly earnings reports, etc.), psychological factors, government involvement and political issues. Therefore, conventional time-series approaches (e.g. AR, ARMA, ARIMA, etc.), based on the linear and stationary figures within a target series, are generally less accurate. Recently, artificial intelligence-based methods such as artificial neural networks (ANNs), support vector regression (SVR),

and fuzzy systems have been successfully applied for stock index forecasting.

ANNs have powerful multi-dimensional non-linear mapping abilities. Moreover, for constructing ANNs, prior domain information is not required and only input-output data pairs are needed. Because of these advantages, ANNs have been widely used for stock index forecasting.

Ticknor [1] proposed a Bayesian regularized ANN to forecast financial market behavior. In [2], the authors proposed the design of a model based on ANN, allowing short term predictions of stock closing prices future behavior. Chang et al. [3] developed an integrated system by combining dynamic time windows, case-based reasoning, and neural networks for stock trading decision support. In [4], a time-series prediction model was developed by combining non-linear independent component analysis (NLICA) and neural networks, which was proposed to forecast Asian stock markets. Bisoi and Dash [5] presented a simple feed-forward dynamic neural network comprising one or more layers of dynamic neurons for predicting stock price indices and profit from one day ahead to 30 days in advance.

If derivative-based optimization methods are employed to train ANNs, there is always the risk of trapping in local minimum. Furthermore, it is difficult to determine the structure (the number of hidden layers and neurons) of ANNs systematically. In other words, ANNs based on the empirical risk minimization principle may suffer from over-fitting problems.

By contrast, SVR, originally introduced by Vapnik, has a global optimum and exhibits better prediction accuracy due to its implementation of the structural risk minimization principle, which considers both the training error and the capacity of the regression model [6]. SVR has been successfully applied in different problems of time-series prediction such as demand forecasting, traffic flow prediction, and financial time series forecasting [7].

Kazem et al. [6] proposed a forecasting model based on SVR, chaotic mapping, and the firefly algorithm to predict stock market prices. Kao et al. [7] developed a stock price forecasting model, NLICA-SVR, which first uses NLICA for pre-processing to extract features from forecasting variables. Xiong et al. [8] investigated the possibility of forecasting an interval-valued stock price index series over short and long horizons using multi-output SVR. Yeh et al. [9] proposed a regression model, which integrates multiple-kernel learning and SVR to deal with the stock price forecasting problem. In [10], a

two-stage architecture for better stock price prediction was employed. The self-organizing map was first used to decompose the whole input space into regions. After decomposing data points into several regions, SVR was applied to forecast stock indices.

To apply SVR to forecasting problems appropriately, the type of kernel functions and associated kernel hyper-parameters must be properly selected in advance. If kernel functions and hyper-parameters are not determined correctly, forecasting performance may be degraded.

Although ANNs and SVR have shown excellent forecasting performance, there are black box problems that these model's operational principles cannot be understood and explained intuitively. Fuzzy systems, on the contrary, are good at explaining their operation principle in the form of readable fuzzy if-then rules [11]. In other words, fuzzy models' operating mechanisms are intuitive and easily understood by designers.

Huang et al. [11] described the application of the hierarchical co-evolutionary fuzzy system for predicting financial time series. To construct an accurate predictive model, a form of generic membership function (MF), named irregular shaped MF, was employed and a hierarchical co-evolutionary genetic algorithm was adopted. Chang and Fan [12] developed a novel approach that integrates the wavelet and TSK fuzzy rule-based systems for stock price prediction. The proposed framework combined several soft computing techniques such as wavelet transform, TSK fuzzy system, data clustering, simulated annealing, and k-nearest neighbor. Hassan et al. [13] proposed a new type of adaptive fuzzy inference system with a view to achieve improved performance for forecasting non-linear time series by dynamically adapting the fuzzy rules with the arrival of new data. Zarandi et al. [14] presented a four-layer fuzzy multi-agent system architecture to develop a hybrid artificial intelligence model based on the coordination of intelligent agents performing data pre-processing and function approximation tasks for next-day stock price prediction.

In this paper, the integration of mutual information (MI) and fuzzy systems for one-day ahead stock index forecasting is proposed. Several technical indicators, widely used in technical analysis, are considered as model's initial input candidates and relevant inputs, highly correlated with model's output, are determined by MI. With the selected inputs, Mamdani-type models which are more intuitive and transparent than TSK-type are constructed. To identify structure and parameters of fuzzy models at the same time, cooperative random learning particle swarm optimization (CRPSO), which was proposed by Zhao et al. [15], is employed. CRPSO is a variant of standard particle swarm optimization (PSO). The structure and parameters of the fuzzy models are encoded into a particle and evolve together so that the optimal structure and parameters can be achieved simultaneously [15]. Extracted fuzzy if-then rules are easily interpreted by human and they might be evaluated by human expertise (rule evaluation). It is also possible to add expert's knowledge-based fuzzy rules to rulebase of fuzzy systems. To verify the effectiveness,

the proposed methods, ANN and SVR are applied to the Korea Composite Stock Price Index (KOSPI) from 1 January 2011 to 31 December 2013. The experimental results show that our method outperforms other comparison methods and the predicted index values by our method are close to the real values.

The remainder of this paper is organized as follows. Section 2 explains the estimation of MI and its use for input selection. Section 3 describes the automatic fuzzy model extraction method using CRPSO. Section 4 shows the experimental results and Section 5 presents concluding remarks and future works.

2. Mutual information for input selection

In this section, input subset selection method for model construction using MI is explained. When conducting system modeling, it is essential to determine significant inputs strongly correlated with output among numerous potential inputs. Using too many inputs lowers model's transparency, increases the number of parameters that must be identified, and raises the risk of over-fitting.

MI measures the information content in input variables with respect to the model output, without making any assumption on the model, and it also has the advantage of detecting non-linear relationships between variables while some other popular criteria such as correlation coefficients are essentially limited to linear dependencies [16], [17].

This study uses a sequential forward input selection method based on MI measurement.

2.1. Definitions of mutual information

Let X and Y be one- or multi-dimensional continuous random variables. If the joint probability density function (PDF) $\mu_{X,Y}$ is given, the marginal PDF of X and Y can be calculated by

$$\mu_X(x) = \int \mu_{X,Y}(x, y) dy \quad (1)$$

and

$$\mu_Y(y) = \int \mu_{X,Y}(x, y) dx. \quad (2)$$

By Shannon's information theory, the uncertainty on Y is calculated by its entropy defined as

$$H(Y) = -\int \mu_Y(y) \log \mu_Y(y) dy. \quad (3)$$

If we get the information about Y from knowing X , the uncertainty on Y is calculated by conditional entropy defined as

$$H(Y | X) = -\int \mu_X(x) \int \mu_Y(y | X = x) \log \mu_Y(y | X = x) dy dx. \quad (4)$$

The joint uncertainty of the (X, Y) pair is calculated by joint entropy defined as

$$H(X, Y) = -\int \mu_{X,Y}(x, y) \log \mu_{X,Y}(x, y) dx dy. \quad (5)$$

The MI between X and Y can be defined by (6), and can be considered as a measure of the amount of knowledge on Y provided by X (or conversely on the amount of knowledge on X provided by Y) [17].

$$I(X, Y) = H(Y) - H(Y|X). \quad (6)$$

As we can see from the above equation, the MI is the decrease of the uncertainty on Y once we know X [18]. By the properties of the MI and entropy, eq. (6) can be rewritten as

$$I(X, Y) = H(X) + H(Y) - H(X, Y) \quad (7)$$

and

$$I(X, Y) = \int \mu_{X,Y}(x, y) \log \frac{\mu_{X,Y}(x, y)}{\mu_X(x)\mu_Y(y)} dx dy. \quad (8)$$

Therefore, in order to compute MI between X and Y , joint PDF between X and Y should be estimated.

2.2. Estimation of the mutual information using k -nearest neighbors

To estimate the joint PDF $\mu_{X,Y}$, histogram-based or kernel-based PDF estimation method is commonly used. However, their usage is commonly limited to functions of one or two variables because the number of samples needed for PDF estimation increases exponentially with the number of variables [19]. In this study, k -nearest neighbor-based MI estimation method proposed by Kraskov et al. [20] is used.

To estimate MI, input-output data pairs $\{\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, n\}$ of target system should be collected. The collected data pairs $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i)$ are assumed to be independent and identically distributed realizations of a random variable $Z = (X, Y)$. The maximum norm used for comparison of any pairs $\mathbf{z} = (\mathbf{x}, \mathbf{y})$ and $\mathbf{z}' = (\mathbf{x}', \mathbf{y}')$ is defined by

$$\|\mathbf{z} - \mathbf{z}'\| = \max(\|\mathbf{x} - \mathbf{x}'\|, \|\mathbf{y} - \mathbf{y}'\|). \quad (9)$$

The basic idea is to estimate $I(X, Y)$ from the distances in spaces X , Y and Z from \mathbf{z}_i to its k -nearest neighbors, averaged over all \mathbf{z}_i [19]. Let us define $\mathbf{z}_{k(i)} = (\mathbf{x}_{k(i)}, \mathbf{y}_{k(i)})$ the k th nearest neighbor of \mathbf{z}_i , where $\mathbf{x}_{k(i)}$ and $\mathbf{y}_{k(i)}$, input and output parts of $\mathbf{z}_{k(i)}$, do not need to be k th nearest neighbor of \mathbf{x}_i and \mathbf{y}_i . Let us denote

$$d_i^X = \|\mathbf{x}_i - \mathbf{x}_{k(i)}\|, d_i^Y = \|\mathbf{y}_i - \mathbf{y}_{k(i)}\| \quad (10)$$

$$\text{and } d_i^Z = \|\mathbf{z}_i - \mathbf{z}_{k(i)}\|.$$

Obviously, $d_i^Z = \max(d_i^X, d_i^Y)$. Sequentially, the number n_i^X of points, whose distance from \mathbf{x}_i is strictly

less than d_i^Z is counted, and similarly the number n_i^Y of points, whose distance from \mathbf{y}_i is strictly less than d_i^Z is counted. In [20], it was proved that MI can be accurately estimated by

$$\hat{I}(X, Y) = \psi(k) - \frac{1}{N} \sum_{i=1}^N [\psi(n_i^X + 1) + \psi(n_i^Y + 1)] + \psi(N), \quad (11)$$

where ψ is the digamma function defined as

$$\psi(t) = \frac{\Gamma'(t)}{\Gamma(t)} = \frac{d}{dt} \ln \Gamma(t), \quad (12)$$

and

$$\Gamma(t) = \int_0^\infty u^{t-1} e^{-u} du. \quad (13)$$

The digamma function satisfies the recursion equation $\psi(t+1) = \psi(t) + 1/t$ and $\psi(1) = C$ where $C = -0.5772156\dots$ is the Euler-Mascheroni constant.

The value of k directly affects the quality of estimated $\hat{I}(X, Y)$. If the value chosen for k is small, $\hat{I}(X, Y)$ has a large variance and a small bias. If the value chosen for k is large, $\hat{I}(X, Y)$ has a small variance and a large bias. In this study, k will be set as commonly used value, $k = 6$.

2.3. Sequential forward input selection using mutual information

Let $\{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, n\}$ be a collected learning dataset, where $\mathbf{x} = [x_1, x_2, \dots, x_m]^T$ is m -dimensional initial input vector, \mathbf{y} is the desired model output, and n is the number of learning data pairs. Collected learning datasets are used for sequential forward input selection method-based on MI.

Firstly, the input x_{1st} that maximizes the MI value in one-dimensional input space is selected by

$$x_{1st} = \arg \max_{x_j} \{\hat{I}(x_j, y)\}, \quad 1 \leq j \leq m. \quad (14)$$

After then, in two-dimensional input space, $m-1$ input combinations are considered. Second best input x_{2nd} is determined by

$$x_{2nd} = \arg \max_{x_j} \{\hat{I}((x_{1st}, x_j), y)\}, \quad (15)$$

$$1 \leq j \leq m, j \neq 1st.$$

Same procedures are repeated until the following condition is satisfied:

$$\hat{I}((x_{1st}, \dots, x_{kth}), y) \geq \quad (16)$$

$$\hat{I}\left((x_{1st}, \dots, x_{kth}, x_{(k+1)th}), y\right) \quad k = 2, \dots, m-1.$$

If the above condition is satisfied, the expansion of input space is stopped and the k th relevant input x_{kth} is decided by

$$x_{kth} = \arg \max_{x_j} \left\{ \hat{I}\left((x_{1st}, x_{2nd}, \dots, x_{(k-1)th}, x_j), y\right) \right\}, \quad (17)$$

$$1 \leq j \leq m, j \neq \{1st, 2nd, \dots, (k-1)th\}.$$

After finishing above procedures, we can get the best input combination $x_{1st}, x_{2nd}, \dots, x_{kth}$ for prediction modeling.

3. CRPSO-based fuzzy model

3.1. Cooperative random learning particle swarm optimization

CRPSO, which was proposed by Zhao et al., employs several sub-swarms to search the space and the useful information is exchanged among them during the iteration process [15]. In CRPSO, at each iteration, velocity vectors are updated by

$$\mathbf{v}_{i,j}(t+1) = w(t)\mathbf{v}_{i,j}(t) + c_1\mathbf{r}_1(t)[\mathbf{y}_{i,j}(t) - \mathbf{x}_{i,j}(t)] + c_2\mathbf{r}_2(t)[\hat{\mathbf{y}}_j(t) - \mathbf{x}_{i,j}(t)] + c_3\mathbf{r}_3(t)[\hat{\mathbf{y}}_r(t) - \mathbf{x}_{i,j}(t)], \quad (18)$$

where $\mathbf{v}_{i,j}(t)$ and $\mathbf{v}_{i,j}(t+1)$ is the velocity vector of i -th particle ($i = 1, \dots, n$) of j -th sub-swarm ($j = 1, \dots, m$) at iteration t and $t+1$, respectively, $w(t)$ is the inertia weight, c_1 , c_2 and c_3 are acceleration constants, $\mathbf{r}_1(t)$, $\mathbf{r}_2(t)$ and $\mathbf{r}_3(t)$ are uniform vector random variables at the range of $[0, 1]$ at iteration t , $\mathbf{x}_{i,j}(t)$ is the position vector of i -th particle of j -th sub-swarm at iteration t , $\mathbf{y}_{i,j}(t)$ is the best position that i -th particle of j -th sub-swarm has visited until now, $\hat{\mathbf{y}}_j(t)$ is the best position that entire particles of j -th sub-swarm have visited until now, and $\hat{\mathbf{y}}_r(t)$ is the randomly selected best position among $\hat{\mathbf{y}}_j(t)$. For further details of standard PSO and CRPSO, see references [15], [21], [22].

3.2. Mamdani-type fuzzy model

The typical multi-input single-output Mamdani-type fuzzy model is composed of a set of fuzzy if-then rules and each rule, which consists of an antecedent part and a consequent part, is expressed by

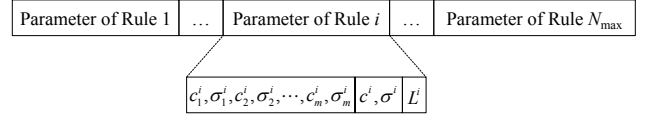


Fig. 1: Coding scheme for extraction of Mamdani-type fuzzy model.

R^i : If x_1 is A_1^i and x_2 is A_2^i and \dots and x_m is A_m^i
Then y is B^i , $i = 1, \dots, N_{\max}$

where x_j ($j = 1, \dots, m$) and y are crisp input and output variables, respectively, A_j^i and B^i are linguistic values defined by fuzzy sets, and N_{\max} is the maximum number of fuzzy rules. The MF of A_j^i is a Gaussian MF defined as

$$\mu_{A_j^i}(x_j; c_j^i, \sigma_j^i) = \exp\left\{-\frac{(x_j - c_j^i)^2}{2(\sigma_j^i)^2}\right\}, \quad (19)$$

where c_j^i and σ_j^i , premise parameters, correspond to the center and the width of the MF, respectively. The output MF, $\mu_{B^i}(y)$, is also a Gaussian MF. The output of Mamdani-type fuzzy model is computed by the centroid of area defined as

$$\hat{y} = \frac{\int_y y \mu_{B^i}(y) dy}{\int_y \mu_{B^i}(y) dy}, \quad (20)$$

where $\mu_{B^i}(y)$ is the aggregated output MF, which is defined as

$$\mu_{B^i}(y) = \max_{i=1, \dots, N_{\max}} \left\{ \min \left[\mu_{A_1^i}(x_1), \mu_{A_2^i}(x_2), \dots, \mu_{A_m^i}(x_m), \mu_{B^i}(y) \right] \right\}. \quad (21)$$

3.3. Encoding scheme of fuzzy systems

Generally, two successive tasks are performed for fuzzy model construction: structure identification and parameter optimization. In the structure identification step, fuzzy rules are generated and boundaries of MF parameters' search space are fixed, and clustering algorithms (e.g. k-means, fuzzy c-means, etc.) have been commonly used. Parameter optimization is to optimize premise and consequent parameters of fuzzy model and derivative-based and derivative-free algorithms (e.g. steepest

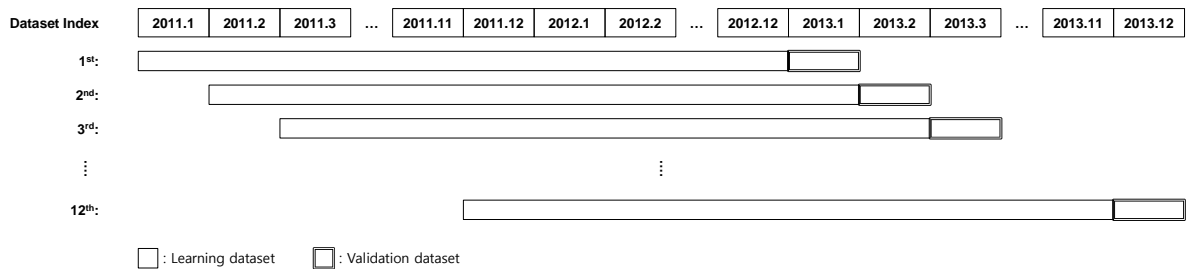


Fig. 2: Preparation of learning and validation dataset (walk-forward testing)

descent, genetic algorithm, simulated annealing etc.) could be employed.

In this study, structure and parameter identification are carried out simultaneously by CRPSO. Fig. 1 shows coding scheme for simultaneous structure and parameters optimization process. As shown in Fig. 1, premise and consequent parameters and labels are encoded into a particle. If label L^i is bigger than zero, then i -th fuzzy rule is activated. Otherwise, the rule is discarded. All the activated fuzzy rules are included in rulebase and used for inference.

4. Experimental results

4.1. Data preparation

In this research, KOSPI time-series data from 1 January 2011 to 31 December 2013 is used to verify the performance of MI-based fuzzy models. Fig. 2 shows the separation scheme of the learning and validation datasets, and similar ways (walk-forward testing) were also used in [23]. Table 1 lists technical indicators widely used in technical analysis of stock markets, and Table 2 sum-

marizes the parameter settings of technical indicators.

The format of initial input candidates and output of learning dataset is $[x_1, x_2, \dots, x_{14}, y]$, where x_1, x_2, \dots, x_{14} are technical indicators listed in Table 1 and y is $C(t+1)$, closing price at time $t+1$. Before selecting relevant inputs explained in Section 2, all input and output variables are normalized at the range of $[0, 1]$ as

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \quad (22)$$

where x is the original value of input or output variable, x_{\max} and x_{\min} are the maximum and minimum values of the original value, respectively, and x_{norm} is the normalized value.

4.2. Model construction

As explained in Section 2.3, relevant inputs, $x_{1st}, x_{2nd}, \dots, x_{nth}$, used for prediction model construction are selected among 14 initial input candidates, x_1, x_2, \dots, x_{14} . Normalized learning datasets are used to calculate MI values. Fig. 3 shows the results of applying MI-based se-

Table 1: Technical indicators used as initial input candidates.

	Name of indicators	Formulas
x_1	Simple moving average (SMA), $SMA_n(t)$	$SMA_n(t) = \frac{C(t) + C(t-1) + \dots + C(t-n+1)}{n} = \frac{1}{n} \sum_{i=1}^n C(t-i+1)$
x_2	Weighted moving average (WMA), $WMA_n(t)$	$WMA_n(t) = \frac{nC(t) + (n-1)C(t-1) + \dots + C(t-n+1)}{n + (n-1) + \dots + 1}$
x_3	Exponential moving average (EMA), $EMA_n(t)$	$EMA_n(1) = C(1)$, $EMA_n(t) = \alpha C(t) + (1-\alpha)EMA_n(t-1)$, for $t > 1$
x_4	Momentum, $Mom_n(t)$	$C(t) - C(t-n+1)$
x_5	Stochastic %K, $StoK_n(t)$	$StoK_n(t) = \frac{C(t) - LL_n(t)}{HH_n(t) - LL_n(t)} \times 100$
x_6	Stochastic %D, $StoD_m(t)$	$StoD_m(t) = \text{'SMA, whose window size is } m, \text{ of } StoK_n(t)\text{'}$
x_7	Relative strength index (RSI), $RSI_n(t)$	$RSI_n(t) = 100 - \frac{100}{1 + RS_n(t)}$
x_8	MACD line, $MACDL(t)$	$MACDL(t) = EMA_{12}(t) - EMA_{26}(t)$
x_9	Signal line, $SignalL(t)$	$SignalL(t) = \text{'9-days EMA of the MACD line'}$
x_{10}	William %R, $WillR_n(t)$	$WillR_n(t) = \frac{HH_n(t) - C(t)}{HH_n(t) - LL_n(t)} \times -100$
x_{11}	Commodity channel index (CCI), $CCI_n(t)$	$CCI_n(t) = \frac{TP(t) - SM_n(t)}{0.015 \times MD_n(t)}$
x_{12}	Rate of change (ROC), $ROC_n(t)$	$ROC_n(t) = \frac{C(t) - C(t-n+1)}{C(t-n+1)} \times 100$
x_{13}	Volume rate of change (VROC), $VROC_n(t)$	$VROC_n(t) = \frac{V(t) - V(t-n+1)}{V(t-n+1)} \times 100$
x_{14}	Accumulation/Distribution line (A/D line), $ADL(t)$	$ADL(1) = MFV(1)$, $ADL(t) = ADL(t-1) + MFV(t)$, for $t > 1$

where $C(t)$, $L(t)$, $H(t)$ and $V(t)$ are the closing price, low price, high price, and volume at time t , respectively, α is a smoothing factor which is commonly calculated by $\alpha = \frac{2}{n+1}$, $LL_n(t) = \min\{L(t), L(t-1), \dots, L(t-n+1)\}$, $HH_n(t) = \max\{H(t), H(t-1), \dots, H(t-n+1)\}$, $RS_n(t)$ is the ratio of total (or average) gains to total (or average) losses in the last n days defined by $RS_n(t) = \frac{\sum_{i=1}^n \text{Gain}(t-i+1)}{\sum_{i=1}^n \text{Loss}(t-i+1)}$, $\text{Gain}(t) = \begin{cases} C(t) - C(t-1), & \text{if } C(t) > C(t-1) \\ 0, & \text{otherwise} \end{cases}$, $\text{Loss}(t) = \begin{cases} C(t) - C(t-1), & \text{if } C(t) < C(t-1) \\ 0, & \text{otherwise} \end{cases}$, $TP(t)$ is the typical price calculated by $TP(t) = \frac{C(t) + H(t) + L(t)}{3}$, $SM_n(t) = \sum_{i=1}^n TP(t-i+1) / n$, $MD_n(t) = \sum_{i=1}^n |TP(t-i+1) - SM_n(t)| / n$, $MFV(t)$ is money flow volume calculated by $MFV(t) = \text{MFM}(t) \times V(t)$, $\text{MFM}(t)$ is money flow multiplier defined by $\text{MFM}(t) = \frac{(C(t)-L(t))-(H(t)-C(t))}{H(t)-L(t)}$.

Table 2: Parameter settings of technical indicators.

Technical indicator	Parameter setting
SMA _n (t), WMA _n (t), EMA _n (t) and Mom _n (t)	n = 10
StoK _n (t) and StoD _m (t)	n = 10 and m = 3
RSI _n (t) and WillR _n (t)	n = 14
CCI _n (t)	n = 20
ROC _n (t) and VROC _n (t)	n = 12

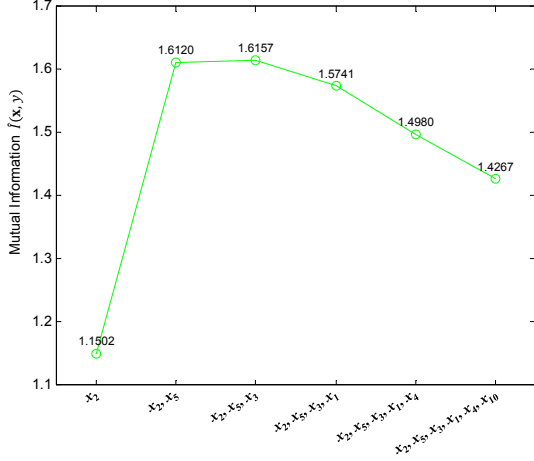


Fig. 3. $\arg \max_{\mathbf{x}_i} \{ \hat{J}(\mathbf{x}_i, y) \}$ vs. $\max_i \{ \hat{J}(\mathbf{x}_i, y) \}$ in 1st dataset.

quential forward input selection method to 1st dataset in Fig. 1. As we can see from Fig. 3, in this case, the expansion of input space is halted at 3-dimension and x_2 , x_5 , and x_3 are sequentially selected. In other words, x_2 , x_5 , and x_3 are used to learn the model for 1st dataset.

After selecting best input combinations, CRPSO-based Mamdani-type fuzzy model extraction is performed as described in Section 3. The parameters of CRPSO are set as follows: the maximum generation number is set as 1000, the number of sub-swarm is set as 3, population size of each sub-swarm is set as 30, the acceleration constants c_1 is set as 1.5, c_2 and c_3 are set as 0.75, $w(t)$ decreases linearly from 0.9 to 0.4 during the optimization process. Fig. 4 shows the example of the convergence behavior of CRPSO.

The same procedures, as explained in Section 4.2, are applied to each prepared dataset in Fig. 1 and the results are presented at Section 4.3.

4.3. Prediction results

The mean absolute percentage error (MAPE) is used as

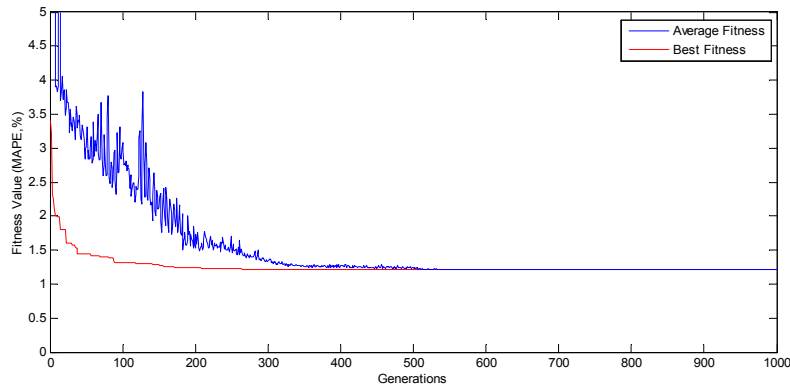


Fig. 4: Example of convergence behavior.

a forecasting performance index. MAPE is defined as

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i} \times 100, \quad (23)$$

where y_i and \hat{y}_i are the actual and predicted output values on i -th validation dataset and N is the number of validation data pairs. Table 3 lists the forecasting results of the proposed method, ANN and SVR in each dataset.

Except for SVR model, in other models (fuzzy and ANN), each experiment is repeated 30 times in the same experimental environments and averaged MAPE values are listed. In Table 3, the best cases among 30 experiments are given in parenthesis. ANN and SVR models are implemented using the MATLAB Neural Network Toolbox and the LIBSVM [24], respectively. In ANN, three layer feed-forward structure is employed, the number of the hidden layer's neurons is fixed as 4, and Levenberg-Marquardt method is used for learning. In SVR, the radial basis function (RBF) kernel is used and hyper-parameters are determined by trial-and-error. As listed in Table 3, the proposed fuzzy model outperforms the comparison methods, SVR and ANN. The proposed method improves the averaged MAPE over SVR and ANN by 0.038% and 0.028%, respectively. Fig. 5 shows the prediction results of the best fuzzy models among 30 repeated experiments and Fig. 6 shows the fuzzy if-then rules, input and output MF of the best extracted fuzzy model in 2nd dataset. As a result, the proposed CRPSO-based fuzzy model shows good non-linear mapping abilities, and is more transparent than ANN and SVR.

5. Conclusions and future works

In this research, the integration of MI and CRPSO-based Mamdani-type fuzzy model was used to construct one-day ahead stock index forecasting model. Technical indicators commonly used in technical analysis were considered as initial input candidates, and significant indicators were selected by the MI-based sequential forward selection method. With the selected indicators, structure and parameter identifications were performed simultaneously by CRPSO. To investigate the effectiveness, the proposed method, ANN and SVR were applied to KOSPI time-series dataset. As presented in Section 4, the proposed method outperformed on average against the other comparison methods. The extract-

Table 3: Prediction results of proposed and comparison methods in 2013 (MAPE).

	Fuzzy		SVR	ANN
	Number of rules	MAPE, %		
Jan.	3.9000 (5)	0.5110 (0.4565)	0.5230	0.5777 (0.4490)
Feb.	3.6000 (3)	0.6517 (0.5875)	0.6886	0.7621 (0.6551)
Mar.	4.0667 (5)	0.7556 (0.4808)	0.6125	0.6994 (0.5495)
Apr.	3.9000 (4)	0.8521 (0.7501)	0.8731	0.9063 (0.7908)
May	3.8000 (5)	0.6071 (0.4266)	0.5642	0.5849 (0.4544)
Jun.	4.1000 (4)	1.1656 (0.9809)	1.2832	1.2686 (1.0173)
Jul.	3.8333 (5)	0.7679 (0.6339)	1.0066	0.7928 (0.6298)
Aug.	3.5333 (5)	0.8975 (0.8223)	0.9830	0.8372 (0.7664)
Sept.	3.7333 (3)	0.5877 (0.4950)	0.7880	0.5504 (0.4811)
Oct.	3.7000 (4)	0.8813 (0.7280)	0.8427	1.0571 (0.5788)
Nov.	3.8667 (5)	0.6764 (0.6076)	0.6066	0.6309 (0.5428)
Dec.	3.6000 (5)	0.5341 (0.4272)	0.5669	0.5594 (0.5047)
Ave.	-	0.7407 (0.6164)	0.7782	0.7689 (0.6183)

ed model structures composed of fuzzy if-then rules are more intuitive and transparent than black box models. Besides, the addition of domain expert's knowledge-based fuzzy if-then rules on rulebase is also possible.

This paper only focused on one-day ahead stock index prediction and considered technical indicators as model's inputs. In further studies, we will expand our method into long-term (e.g. 5days, 10days, 20days, etc.) prediction models and consider possibilities of employing economic exogeneous variables (e.g. interest rate, exchange rate, international oil and gold price, etc.), that might influence stock index movements, as input candidates of prediction models.

Acknowledgment

This research was supported by the MOTIE (Ministry of Trade, Industry & Energy), Korea, under the Industry Convergence Liaison Robotics Creative Graduates Ed-

ucation Program supervised by the KIAT (N0001126).

References

- [1] J. L. Ticknor, A Bayesian regularized artificial neural network for stock market forecasting, *Expert Systems with Applications*, 40:5501-5506, 2013.
- [2] F. A. de Oliveira, C. N. Nobre and L. E. Zárate, Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index - Case study of PETR4, Petrobras, Brazil, *Expert Systems with Applications*, 40:7596-7606, 2013.
- [3] P. C. Chang, et al., A neural network with a case based dynamic window for stock trading prediction, *Expert Systems with Applications*, 36: 6889-6898, 2009.
- [4] W. Dai, J. Y. Wu and C. J. Lu, Combining nonlinear independent component analysis and neural network for the prediction of Asian stock market indexes, *Expert Systems with Applications*, 39: 4444-4452, 2012.
- [5] R. Bisoi and P. K. Dash, A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter, *Applied Soft Computing*, 19: 41-56, 2014.
- [6] A. Kazem, et al., Support vector regression with chaos-based firefly algorithm for stock market price forecasting, *Applied Soft Computing*, 13: 947-958, 2013.
- [7] L. J. Kao, et al., Integration of nonlinear independent component analysis and support vector regression for stock price forecasting, *Neurocomputing*, 99: 534-542, 2013.
- [8] T. Xiong, Y. Bao and Z. Hu, Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting, *Knowledge-Based Systems*, 55: 87-100, 2014.
- [9] C. Y. Yeh, C. W. Huang and S. J. Lee, A multiple-kernel support vector regression approach for stock market price forecasting, *Expert Systems with Applications*, 38: 2177-2186, 2011.
- [10] S. H. Hsu, et al., A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression, *Expert Systems with Applications*, 36: 7947-7951,

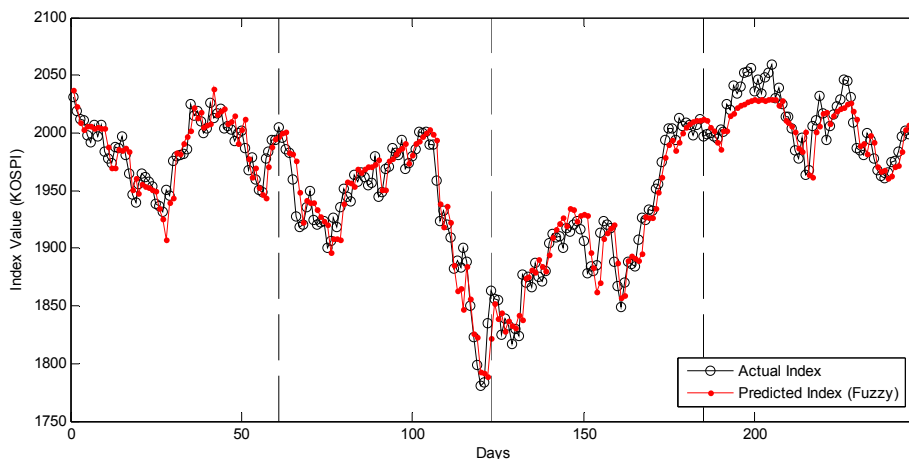


Fig. 5: Best prediction results of fuzzy model (KOSPI, 2013).

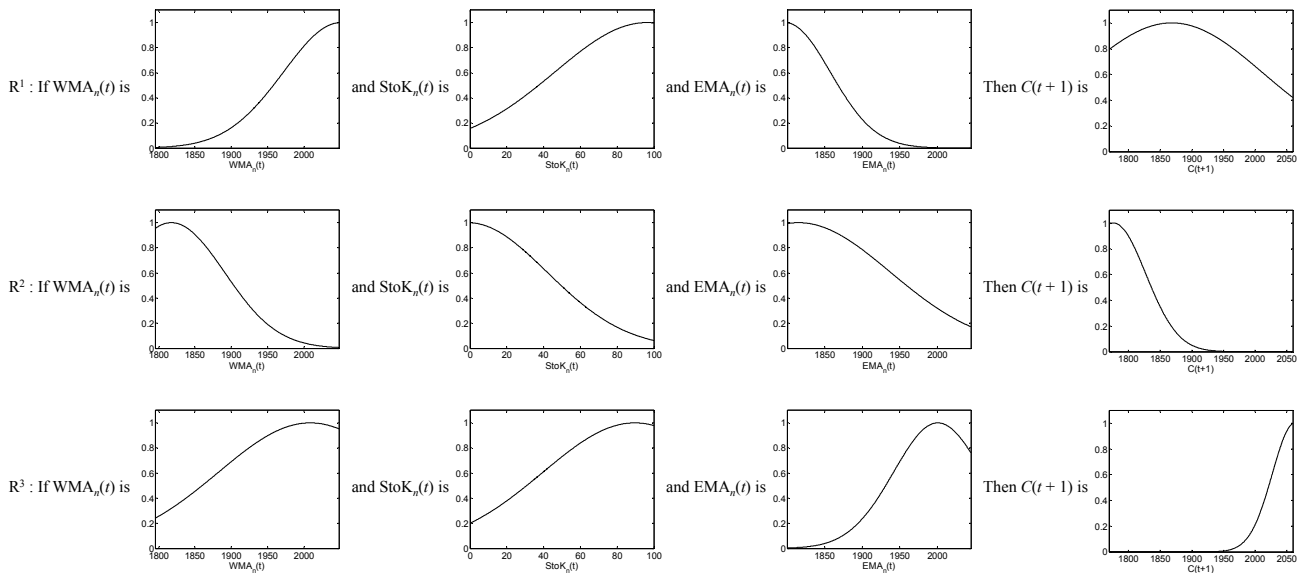


Fig. 6: Best extracted fuzzy model in 2nd dataset.

2009.

- [11] H. Huang, M. Pasquier and C. Quek, Financial market trading system with a hierarchical coevolutionary fuzzy predictive model, *IEEE Transactions on Evolutionary Computation*, 13: 56-70, 2009.
- [12] P. C. Chang and C. Y. Fan, A hybrid system integrating a wavelet and TSK fuzzy rules for stock price forecasting, *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 38: 802-815, 2008.
- [13] M. R. Hassan, et al., A HMM-based adaptive fuzzy inference system for stock market forecasting, *Neurocomputing*, 104: 10-25, 2013.
- [14] M. H. Zarandi, E. Hadavandi and I. B. Turksen, A hybrid fuzzy intelligent agent-based system for stock price prediction, *International Journal of Intelligent Systems*, 27: 947-969, 2012.
- [15] L. Zhao, et al., Automatically extracting T-S fuzzy models using cooperative random learning particle swarm optimization, *Applied Soft Computing*, 10: 938-944, 2010.
- [16] B. Frénay, G. Doquire and M. Verleysen, Is mutual information adequate for feature selection in regression?, *Neural Networks*, 48: 1-7, 2013.
- [17] F. Rossi, et al., Mutual information for the selection of relevant variables in spectrometric nonlinear modelling, *Chemometrics and Intelligent Laboratory Systems*, 80: 215-226, 2006.
- [18] A. Guillen, et al., New method for instance or prototype selection using mutual information in time series prediction, *Neurocomputing*, 73: 2030-2038, 2010.
- [19] M. B. Stojanović, et al., A methodology for training set instance selection using mutual information in time series prediction, *Neurocomputing*, 141: 236-245, 2014.
- [20] A. Kraskov, H. Stögbauer and P. Grassberger, *Estimating mutual information*, Physical review, 69: 066138, 2004.
- [21] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, 4: 1942-1948, 1995.
- [22] A. Engelbrecht, *Computational Intelligence: An introduction*, 2nd ed. NY, USA: John Wiley & Sons, 2007.
- [23] L. A. Teixeira and A. L. I. de Oliveira, A method for automatic stock trading combining technical analysis and nearest neighbor classification, *Expert systems with applications*, 37: 6885-6890, 2010.
- [24] C. C. Chang and C. J. Lin, LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology*, 2: 27:1-27:27, 2011.