

Article

Incorporating a New Weighting Scheme into Decomposition Ensemble Models for Forecasting Air Passenger Flow by Combining Fuzzy Cognitive Maps with Grey Relational Analysis

Yi-Chung Hu ^{1,*}, Geng Wu ² and Yu-Chao Cheng ³¹ Department of Business Administration, Chung Yuan Christian University, Taoyuan City 320314, Taiwan² School of Economics and Management, Ningbo University of Technology, Ningbo 315211, China; wugeng@nbut.edu.cn³ School of Business, Singapore University of Social Sciences, Singapore 599494, Singapore; jackchengyc@suss.edu.sg

* Correspondence: ychu@cycu.edu.tw

Abstract

Previous studies in passenger flow forecasting commonly employ decomposition ensemble models with linear addition, treating individual single-component forecasts with equal weights, to produce ensemble forecasts. It is known that fuzzy cognitive maps (FCMs) in multiple criteria decision-making are capable of modeling system dynamics by catching the causal relationships between the concepts describing a system. To enhance the forecasting ability of decomposition ensemble models with linear addition, this study aims to develop weighting schemes based on FCMs and grey relational analysis (GRA). Time series are decomposed into several components, and neural networks are applied to forecast individual components. Then, GRA is applied to assess the weights for individual single-component forecasts. To obtain ensemble forecasts, an optimal FCM determined by a genetic algorithm is used to determine the final combination weights for individual single-component forecasts. In comparison with benchmark models, the results show that the proposed FCM-based decomposition ensemble models significantly effectively improve the forecasting accuracy of air passenger flow in Taiwan across different forecasting horizons.

Keywords: grey relational analysis; fuzzy cognitive map; ensemble learning; passenger flow; MCDM

MSC: 68T20

Academic Editor: Vladimir Balan

Received: 25 June 2026

Revised: 31 July 2026

Accepted: 2 August 2026

Published: 4 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Forecasting demand in any industry is increasingly important to confront external competition and enhance operational performance [1,2]. The accuracy of forecasts of the flow of air passengers has received considerable attention from researchers and industry practitioners in recent years [3]. Such forecasts are advantageous to both policymakers and air transport practitioners to formulate management strategies and operational plans [4,5].

For time series modeling, decomposition ensemble learning has attracted extensive attention from researchers for forecasting passenger demand, given that it may improve the forecasting accuracy of single models [3]. The construction of a decomposition ensemble model comprises three main stages: data decomposition, component series forecasting, and ensemble forecast generation [6,7]. Empirical evidence has demonstrated the superiority of decomposition ensemble models over single forecasting models, including those such as Bi et al. [6], Hu et al. [8], Tang et al. [9], Xing et al. [10], and Liang et al. [11], among others. Even so, further improvements to the forecasting performance of decomposition ensemble models are needed for passenger demand forecasting.

Data decomposition can reduce the complexity of time series by dividing data series into several components with low volatility through various data decomposition methods [12,13]. Some decomposition methods have been utilized to extract meaningful patterns embedded in time series associated with passenger demand to improve the capability of forecasting models, such as empirical mode decomposition (EMD) [8,9,14–18], variational mode decomposition [3,10], seasonal and trend decomposition using locally estimated scatterplot smoothing (STL) [19], and structured time series models (STSMs) [10].

In respect of component series forecasting, because of the capability of neural networks (NNs) for modeling nonlinearity and mapping arbitrary functions [5], NNs were commonly used to forecast component series, such as support vector regression (SVR) by Shi et al. [20], Xie et al. [21], and Xing et al. [10]; echo state network by Qin et al. [19]; long short-term memory (LSTM) by Liang et al. [11], Liu et al. [15], Tang et al. [22], Wu et al. [23], and Yang et al. [24]; radial basis function network (RBFN) by Li et al. [25]; and multi-layer perceptron (MLP) by Zhang et al. [18]. Ensemble forecasts are then produced by fusing single-component forecasts. Linear addition with equal weight that simply adds single-component forecasts together to form ensemble forecasts is frequently applied to forecast passenger flow such as Cao et al. [14], Liang et al. [11], Qin et al. [19], and Zhang et al. [18]. However, while equal weight is simple enough, weighting schemes that are used to assess the contribution of partial performance to the synthetic results can have a great effect on the accuracy [8].

Here, we focus on the weighting scheme with respect to linear addition for decomposition ensemble modeling. First, the use of the correlation between components and the original series is beneficial for identifying influential components for ensemble learning [16]. It is reasonable to treat air passenger demand as a grey system. Although series of passenger flow data are available for analyzing air passenger demand, the factors that can affect demand are not completely known. This motivates us to exploit grey relational analysis (GRA), an effective tool commonly used for grey systems analysis [26–29], to identify the weights of individual single-component forecasts by assessing the correlation among series. Also, little attention has been paid to applying the perspective of grey systems to forecast the demand for passengers in transportation.

Furthermore, because the sources of information are not always independent of one another [30], it is feasible to determine the weights of individual single-component forecasts by identifying the interrelationships among component series forecasting models. This inspired us to incorporate fuzzy cognitive maps (FCMs), as proposed by Kosko [31], into the weighting scheme. FCMs are characterized by their ability to model the dynamics of complex systems by identifying the causal relationships among factors [32,33]. FCMs can thus generate final solutions in response to a given set of initial conditions. They are applied in diverse fields such as pattern recognition [34–36], time series forecasting [37–40], policy formulation [41,42], and industrial applications [33,43]. To our knowledge, despite the usefulness of FCMs, no study has integrated them with decomposition ensemble modeling in the field of transportation.

In sum, this study features a weighting scheme for linear addition: Given state values pertaining to the initial weights of individually forecasted components through GRA, FCMs are utilized to determine the final solution by identifying the interrelationships among component series forecasting models. With data series decomposition, component series forecasting, and ensemble forecast generation for forecasting air passenger flow, we aimed to examine whether incorporating FCM-based weighting schemes into linear addition improves the accuracy of decomposition ensemble models. The forecasting performance of the proposed decomposition ensemble models was examined using the flow of air passengers to Taiwan from six primary airports. Overall, the contributions of this study are as follows:

- This study presented a new weighting scheme for decomposition ensemble learning to improve the accuracy of air passenger demand forecasting.
- The results showed that the proposed methods with GRA and FCM outperformed single forecasting models and other benchmark models considered.
- To our knowledge, despite the usefulness of FCMs, no study has integrated them with decomposition ensemble modeling in the field of transportation.

The remainder of this paper is organized as follows. Section 2 describes the methodological framework, Section 3 introduces the proposed FCM-based weighting schemes and the resulting decomposition ensemble model, and Section 4 compares its accuracy to that of several benchmark models based on statistical tests. Conclusions are presented in Section 5.

2. Methodology

A three-stage methodological framework is depicted in Figure 1. The first stage involves dividing the original data series into multiple components by EMD. This approach has been previously successful in travel and passenger demand forecasting. In the second stage, commonly used neural networks, including the MLP, the RBFN, SVR, convolutional neural networks (CNNs), and LSTM, are exploited to generate single-component forecasts. GRA is applied to assess the state values associated with the initial weights of individual component series forecasting models, and FCMs with a learning algorithm evolve to reach a steady state with respect to the final solution. Then, linear addition synthesizes single-model forecasts into composite forecasts. We examined the forecasting performance of the proposed decomposition ensemble models using the flow of air passengers to Taiwan from six primary airports.

2.1. Empirical Mode Decomposition

EMD was developed on the assumption that any non-stationary and nonlinear time series is composed of different simple intrinsic modes of oscillation expressed by intrinsic mode functions (IMFs) [44]. Some studies combined EMD with NNs to develop hybrid methods to obtain better forecasting performance [15]. EMD applies a shifting process to decompose the original series y into z IMFs and a residual such that $y = (y(1), y(2), \dots, y(n))$ can be represented as

$$y(t) = \sum_{i=1}^z IMF_i(t) + r_z(t) \quad (1)$$

where $IMF_i(t)$ and $r_z(t)$ denote the i -th IMF and the residue at time t , respectively. Initially, let s , k , $r_0(t)$, and $h_1(t)$ equal 1, 1, $y(t)$, and $y(t)$, respectively. To extract the s -th IMF, the first step of the decomposition procedure identifies all local extrema, including the maximum and minimum values of $h_k(t)$. With the interpolation of two cubic splines, we generate the upper ($u_k(t)$) and lower ($l_k(t)$) envelopes defined by the maxima and the minima, respectively. Then, a mean envelope $m_k(t)$ is computed as

$$m_k(t) = \frac{u_k(t) + l_k(t)}{2} \quad (2)$$

The extracted signal $h_{k+1}(t)$ is the difference between $h_k(t)$ and $m_k(t)$,

$$h_{k+1}(t) = h_k(t) - m_k(t) \quad (3)$$

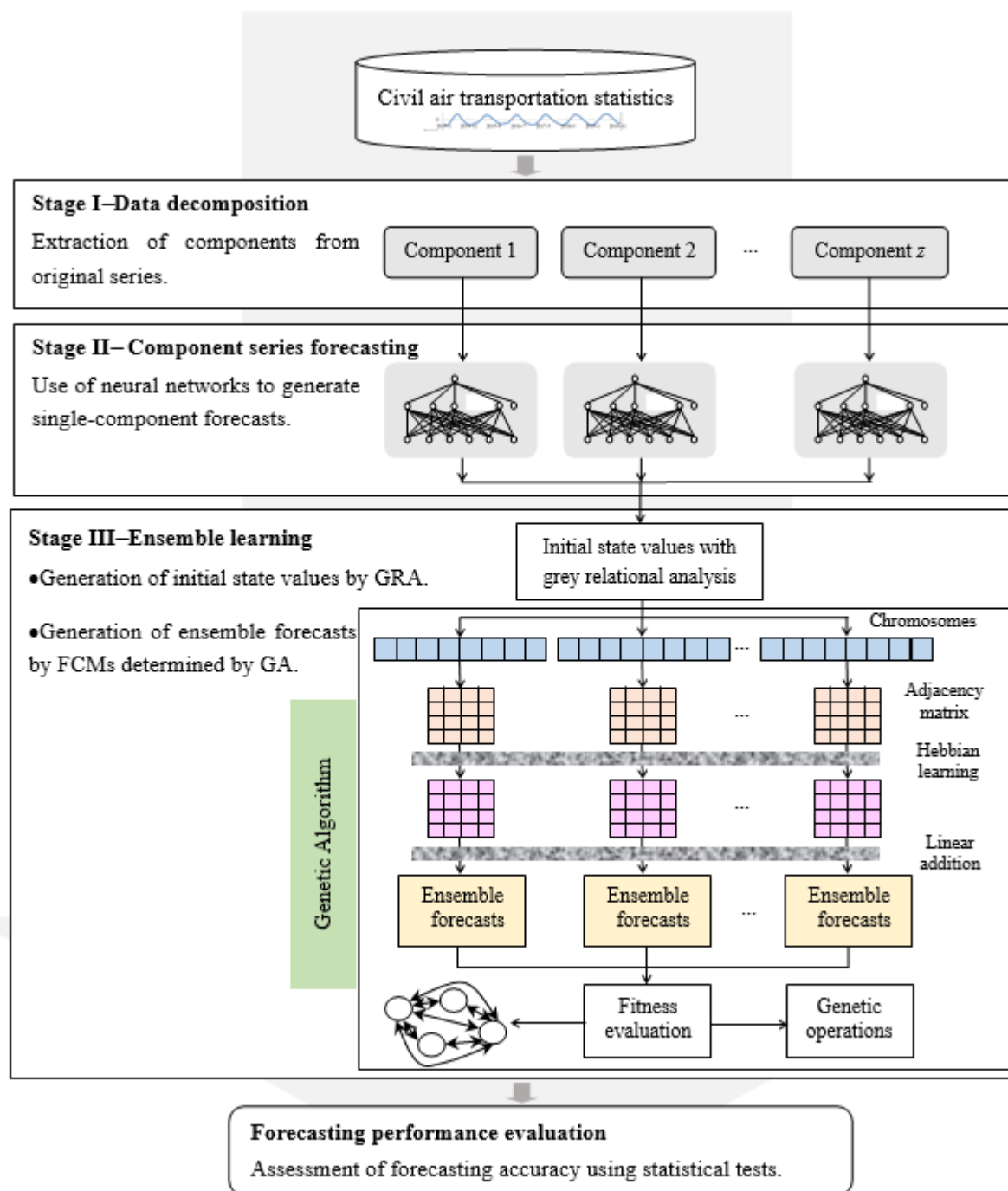


Figure 1. The methodological framework.

At this stage, it is necessary to check whether $h_{k+1}(t)$ accords with the conditions with respect to the IMF: (1) at any point, the means of the local upper envelope and the lower envelope are equal to or close to zero; and (2) the number of extreme points and zero crossing points must be either equal or differ at most by 1. If $h_{k+1}(t)$ does not satisfy these conditions, then $h_k(t)$ is replaced by $h_{k+1}(t)$, k is increased by 1, and the above procedure is

iterated until the s -th IMF is generated. Otherwise, define $\text{IMF}_s(t) = h_{k+1}(t)$, the residual $r_s(t)$ is computed as

$$r_s(t) = r_{s-1}(t) - h_{k+1}(t) \quad (4)$$

If $r_s(t)$ represents a residue, then the process is terminated; otherwise, we return to the first step of the decomposition procedure and the procedure is repeated until the residue is stationary by increasing s by 1, and setting k and $h_1(t)$ as 1 and $r_{s-1}(t)$, respectively, where $r_z(t)$ is the last IMF extracted from data series.

2.2. Neural Networks

NNs like SVR, RBFN, MLP, CNN, and LSTM have been widely used to forecast passenger demand. The proposed model thus applies these NNs to single-component forecasting. Let $X_i(t) = \{y(t-1), y(t-2), \dots, y(t-\beta)\}$ denote the input space at time t ($t = \beta + 1, \beta + 2, \dots, n$), where β denotes the lag order, while $\hat{y}_i(t)$ representing the single-component forecast obtained from the i -th component series forecasting model i ($1 \leq i \leq z$) is the corresponding 1-step-ahead forecast. With the kernel function $(X_i(t), X(t))$, SVR generates $\hat{y}_i(t)$ as follows:

$$\hat{y}_i(t) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(X_i(t), X(t)) + b \quad (5)$$

where $X(t)$ has the form of $X_i(t)$, and $\alpha_i, \alpha_i^* \in [0, C]$ represent the Lagrange multipliers in a quadratic programming problem. Gaussian or linear functions can serve as the kernel function for SVR. For the RBFN, the activation function of a hidden neuron considers the Gaussian function. With h hidden nodes, let σ_k and μ_k in the Gaussian function represent the spread width and center, respectively, for the k -th hidden node ($1 \leq k \leq h$). The corresponding output can be written as

$$R_k(X(t)) = \exp\left(-\frac{\|X(t) - \mu_k\|^2}{2\sigma_k^2}\right) \quad (6)$$

where $\|\cdot\|$ denotes the Euclidean norm. As a result, $\hat{y}_i(t)$ produced by the RBFN is written as

$$\hat{y}_i(t) = \sum_{k=1}^h w_k R_k(X(t)) \quad (7)$$

For an MLP, $\hat{y}_i(t)$ is expressed by the following [45]:

$$\hat{y}_i(t) = \sum_{k=1}^h v_k g\left(\sum_{j=1}^p u_{kj} y(t-j)\right) \quad (8)$$

where g is an activation function, u_{kj} denotes the weight of the connection linking the j -th input node to the k -th hidden node, and v_k denotes the weight of the connection linking the k -th hidden node to the output node.

The convolutional layer, pooling layer, and fully connected layer are the three basic layers of a CNN. In the convolutional layer, each filter slides over the input sequence and computes a weighted sum of the local input values. The resulting feature map value can be expressed as follows:

$$F_{ij}^{(l)} = \sigma\left(\sum_m \sum_n w_{mn}^{(l)} x_{(i+m)(j+n)}^{(l-1)} + b^{(l)}\right) \quad (9)$$

where $x_{(i+m)}^{(l-1)}$ denotes the input at position $i + m$, $w_{mn}^{(l)}$ the convolutional filter weight, σ is an activation function, and $b^{(l)}$ denotes a bias term. A down-sampling operation is conducted by the pooling layer to lower the spatial extent, whereas the fully connected layers are designed with a metaheuristic.

LSTM copes with data series at time t as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, X(t)] + b_f) \quad (\text{forget}) \quad (10)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, X(t)] + b_i) \quad (\text{input}) \quad (11)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, X(t)] + b_o) \quad (\text{output}) \quad (12)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (\text{cell state}) \quad (13)$$

$$h_t = o_t \odot \tanh(C_t) \quad (\text{hidden state}) \quad (14)$$

$$\tilde{C}_t = (W_c \cdot [h_{t-1}, X(t)] + b_c) \quad (\text{candidate cell state}) \quad (15)$$

in which f_t , i_t , and o_t denote different gates that control the flow of information, $\odot$ means a Hadamard operation, and $[h_{t-1}, Y_t]$ denotes the concatenation of the previous hidden state and the current input. Further, W_f , W_i , and W_o represent the weight matrices, and b_f , b_i , and b_o represent biases. σ is an activation function.

The NNs considered were all implemented in Python 3.9.25. The scikit-learn 1.4.1 was applied to the construction of the MLP and SVR. The RBFN was built with the PyTorch 2.8.0, and the CNN and LSTM were constructed with the TensorFlow 2.7.0. The searched values for hyperparameters pertaining to a grid search used by these NNs are presented in Table 1. The lag order and hyperparameters were selected primarily according to training set performance, and an independent validation procedure was not implemented. In addition, the implemented CNN consisted of one Conv1D layer, a kernel size of 1, and a ReLU activation function, followed by a max pooling layer with a pool size of 1, a flattening layer, a dropout layer with a dropout rate of 0.3, and a fully connected linear output layer with one node. As for LSTM, it consisted of one LSTM layer with a hyperbolic tangent activation function (tanh), followed by a dropout layer with a dropout rate of 0.5 and a fully connected linear output layer with one node. The loss function, optimizer, batch size, and random seed are MSE, Adam, 16, and 42, respectively, and are specified to implement CNN and LSTM.

Table 1. The searching values in grid search for neural networks.

Model	Parameter	Description	Search Value
All models	lag order	Number of preceding time points used as predictors	1, 2, 3, 4, 5, 6
	kernel	Kernel function	rbf, linear
	C	Regularization parameter. The strength of the regularization is inversely proportional to C	$2^{-10}, 2^{-9}, 2^{-8}, 2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}$
	gamma	Kernel coefficient	$2^{-10}, 2^{-9}, 2^{-8}, 2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}$
RBFN	hidden nodes	The number of hidden nodes	2, 4, 6, 8, 10, 12, 14
MLP	activation	The hidden node's activation function	identity, logistic, tanh, ReLU
	hidden nodes	The number of hidden nodes	2, 4, 6, 8, 10, 12, 14
CNN	learning rate	The value of learning rate	0.0001, 0.001, 0.01, 0.1
	filters	The number of filter maps in the convolutional layer	8, 16, 32, 64, 128, 256
LSTM	learning rate	The value of learning rate	0.0001, 0.001, 0.01, 0.1
	nodes	The number of nodes to use in the hidden layer	8, 16, 32, 64, 128, 256

2.3. Grey Relational Analysis

GRA can identify the degree of similarity in the developing trend among time series in a given system. By treating y as the reference sequence and the i -th forecasted component series $\hat{y}_i = (\hat{y}_i(1), \hat{y}_i(2), \dots, \hat{y}_i(n))$ ($1 \leq i \leq z$) as the comparative sequences, the grey relational coefficient ξ_{ij} between $y(j)$ and $\hat{y}_i(j)$ ($1 \leq j \leq n$) can be computed as follows [26,29]:

$$\xi_{ij} = \frac{\Delta_{\min} + \rho \Delta_{\max}}{\Delta_{ij} + \rho \Delta_{\max}} \quad (16)$$

where ρ represents a discriminative coefficient ($0 \leq \rho \leq 1$), and is usually set to 0.5. Further,

$$\Delta_{\min} = \min_{i=1..z} \min_{j=1..n} |y(j) - \hat{y}_i(j)| \quad (17)$$

$$\Delta_{\max} = \max_{i=1..z} \max_{j=1..n} |y(j) - \hat{y}_i(j)| \quad (18)$$

$$\Delta_{ij} = |y(j) - \hat{y}_i(j)| \quad (19)$$

Obviously, ξ_{ij} approaches one when Δ_{ij} is near $\Delta_{\min}$. The grey relational grade (GRG) representing the overall degree of correlation of IMF_i with respect to the original series is written by the following:

$$r_i = \frac{1}{n} \sum_{j=1}^n \xi_{ij} \quad (20)$$

Because Wei and Chen [16] applied Pearson correlation and Kendall correlation to measure the correlation between each IMF and the original time series without taking the scale of a component into account, we applied GRA without using the normalization to measure the correlation between the original series and IMF_i .

It should be noted that—unlike traditional correlation analyses such as the commonly used Pearson correlation or Kendall correlation, which merely stress the correlation coefficients between two random variables—GRA features the identification of the correlation between the reference sequence and comparative sequences by viewing the reference sequence as a desired goal that each comparative sequence expects to achieve.

2.4. Fuzzy Cognitive Maps

FCM dynamics analysis examines how the causal relationships among nodes influence the temporal evolution of node states and the steady states ultimately reached by the system. The state value of a node can be represented by its relative weight [46]. Therefore, the GRG of each component series forecasting model can be used as the initial state value of the corresponding node. In respect of an FCM, the concepts of a modeled system are represented as nodes, and the nodes are interconnected through directed arcs with weights that denote the causal relationships between nodes. An FCM is denoted as a tuple of (C, A, W) , where $C = \{c_1, c_2, \dots, c_z\}$ denotes the set of nodes representing z component series forecasting models. Further, $A = \{<c_i, c_j> | c_i, c_j \in C\}$ ($1 \leq i, j \leq z$) is the set of the directed arcs, and $W = [w_{ij}]$ is the adjacency matrix, where $w_{ij} \in [-1, 1]$ is the weight of the arc $<c_i, c_j>$, and $|w_{ij}|$ denotes the strength of the effect of c_i on c_j . The sign of w_{ij} indicates the direction of influence: $w_{ij} > 0$ represents that an increase (decrease) in c_i gives rise to an increase (decrease) in c_j , $w_{ij} < 0$ represents that an increase (decrease) in c_i gives rise to a decrease (increase) in c_j , and $w_{ij} = 0$ represents no relationship between c_i and c_j .

Let $s_i(\alpha) \in [0, 1]$ represent the state value of c_i at the α -th iteration ($\alpha \geq 0$) so that $s_1(0), s_2(0), \dots, s_z(0)$ are initial state values of individual single-component forecasts and are

measured by grey relational grades. The state value $s_i(\alpha + 1)$ of c_i is calculated by summarizing the influences of all the other concepts to c_i as follows [46,47]:

$$s_i(\alpha + 1) = f(s_i(\alpha) + \sum_{j=1, j \neq i}^z w_{ji}(\alpha) s_j(\alpha)) \quad (21)$$

α equals 0 initially. A monotonically non-decreasing nonlinear function is employed to squash the activation value of each node to an allowed interval, and the overall impact can be clamped to $[0, 1]$. The sigmoid function can serve as the following threshold function because it exhibits superior forecasting capability [35]:

$$f(x) = \frac{1}{1 + e^{-\omega x}} \quad (22)$$

where ω is the sigmoid slope.

To enhance the effectiveness of the adjacency matrix for time series modeling, W should be tuned using unsupervised Hebbian learning to find the final weight set of an FCM. Given initial state values, all FCM nodes update their state values at each iteration. Then, each interconnection weight is updated using the weight updating rule, and a new weight is obtained for the next iteration. It turns out that W changes along with the FCM dynamics. The frequently used Nonlinear Hebbian learning rule (NHL) is formulated as follows:

$$w_{ij}(\alpha + 1) = w_{ij}(\alpha) + \eta s_j(\alpha)(s_i(\alpha) - s_j(\alpha)w_{ij}(\alpha)) \quad (23)$$

where η is the learning rate belonging to $[0, 0.1]$.

Termination criteria can be either the minimization of the variation in two subsequent state values of each node (criterion I) [46] or the maximization of iterations (criterion II) [33]. The former is expressed by the following:

$$|s_i(\alpha + 1) - s_i(\alpha)| \leq \varepsilon \quad (24)$$

where ε is a small positive number in $[0, 1]$.

2.5. Evaluation of the Forecasting Accuracy

To assess the forecasting performance of different decomposition ensemble methods with linear addition, the m -step-ahead forecast $\hat{y}(t + m)$, which involves relating past observations to the currently forecasted variable, is formulated as follows:

$$\hat{y}(t + m) = h(y(t), y(t - 1), \dots, y(t - (\beta - 1))) \quad (25)$$

where $\hat{y}(t + m)$ is an m -step-ahead forecast at time t ($t + m \leq n$). In general, the lag order can be determined by the autocorrelation function and the partial correlation function. However, to maximize the forecasting accuracy of AI-based methods, trial and error can be used to determine the appropriate structures [5,48]. The best training performance of individual AI-based methods, considered among lag orders ranging from two to 12, was thus determined for each component series. Let $\hat{e}(t)$ denote the ensemble forecast generated at time t . The root mean square error (RMSE) is applied to measure forecasting accuracy:

$$\text{RMSE} = \sqrt{\frac{\sum_{k=1}^n (\hat{e}(k) - y(k))^2}{n}} \quad (26)$$

The smaller the values of the RMSE, the more accurate the forecast of the corresponding method.

3. FCM-Based Decomposition Ensemble Models

To construct the decomposition ensemble models with the weighting scheme based on GRA and FCM (GF), a real-valued genetic algorithm (GA), which is a powerful optimization technique inspired by the principles of natural selection and evolution [49], is devised to determine the optimal sigmoid slope (ω), z adjusted weight coefficients ($\delta_1, \delta_2, \dots, \delta_z$) for adjusting state values, and $z(z-1)$ interconnection weights ($w_{12}, w_{13}, \dots, w_{1z}, w_{21}, w_{23}, \dots, w_{2z}, \dots, w_{z,z-1}$). An FCM corresponding to a chromosome consisting of $z^2 + 1$ ($z(z-1) + z + 1$) genes is created, and the initial state values are presented to the FCM. Then, one Hebbian learning rule is used to tune the adjacency matrix until the termination condition is met. As a result, the final state values serve as the weights for linear addition, and the resulting RMSE is used to measure a chromosome's fitness.

A computer program using Delphi XE2 was used to implement the GA to identify the optimal values of the sigmoid slope and weights pertaining to the FCM. We assigned 500, 200, 0.95, and 0.01 to the maximum number of generations, the population size, crossover rate, and mutation rate, respectively, for the GA. ϵ is specified as 0.01. The GA-based learning algorithm for establishing the proposed FCM-based decomposition ensemble model was devised as follows.

Step 1. Decomposition of time series: EMD is exploited to decompose a training data series into z components.

Step 2. Acquisition of single-component forecasts: z NNs are constructed to yield individual single forecasts.

Step 3. Assessment of interrelationships among series: The initial state value $s_i(0)$ ($1 \leq i \leq z$) is yielded by applying GRA to derive the GRG of the forecasted component series $\hat{y}_i$ with respect to the training data. The greater the GRG of $\hat{y}_i$, the greater degree of correlation between the original series and $\hat{y}_i$.

Step 4. Population initialization: We randomly generate an initial population of chromosomes. Each chromosome corresponding to an FCM comprises the sigmoid slope, $z(z-1)$ interconnection weights and z adjusted weight coefficients. At the outset, $\alpha = 0$ and Steps 5–9 are performed for each chromosome.

Step 5. Computation of new state values: The current state values at time t are presented to an FCM to obtain new state values at iteration $\alpha + 1$.

Step 6. Adjustment of interconnection weights: The interconnection weights in an adjacency matrix are tuned using the NHL.

Step 7. Hebbian learning termination criterion I test: The learning process is terminated when termination criterion I (the variation in two subsequent state values of each node less than ϵ) is satisfied. Then, we go to Step 9.

Step 8. Hebbian learning termination criterion II test: The learning process is terminated when termination criterion II (maximum iterations ρ) is satisfied; otherwise, we assign $\alpha + 1$ to α and return to Step 5.

Step 9. Chromosome evaluation: The final state value $s_k(\alpha)$ ($1 \leq k \leq z$) can be replaced by $\alpha_k s_k(\alpha)$ when the rankings of the elements in the sets $\{\alpha_k s_k(\alpha) | k = 1, \dots, z\}$ are the same as those in $\{s_k(\alpha) | k = 1, \dots, z\}$, where α_k ($0 < \alpha_k < 1$) is called the adjusted weight coefficient [50] with respect to the k -th final state; otherwise, $s_k(\alpha)$ remains unchanged. This means that $s_k(\alpha)$ shrinks by using δ_k , but the ranking of $\delta_1 s_1(\alpha), \delta_2 s_2(\alpha), \dots, \delta_z s_z(\alpha)$ should be consistent with that of $s_1(\alpha), s_2(\alpha), \dots, s_z(\alpha)$. Using linear addition, the ensemble forecast $\hat{y}(t)$ is expressed as follows:

$$\hat{y}(t) = \sum_{i=1}^z s_i(\alpha) \hat{y}_i(t) \quad (27)$$

The fitness value of each chromosome was measured on the basis of the RMSE.

Step 10. GA termination test: The whole algorithm is terminated after creating the maximum number of generations. An optimal FCM can be obtained from the best chromosome whose fitness value is a minimum among the straight generations.

Step 11. Generation of new chromosomes: Once the fitness of each chromosome in the current population has been acquired, for the next population, new chromosomes are produced using genetic operations, including selection, crossover, and mutation.

- (1) Selection: With binary tournament selection with replacement, two strings are randomly selected from the current population, and the string with less fitness is put in the mating pool. This task is carried out repeatedly until the number of newly generated strings equals the population size.
- (2) Crossover and mutation: Two parent strings, say $\omega_u, \omega_{u,12}, \omega_{u,13}, \dots, \omega_{u,1z}, \omega_{u,21}, \omega_{u,23}, \dots, \omega_{u,2z}, \dots, \omega_{u,zz-1}$ and $\omega_v, \omega_{v,12}, \omega_{v,13}, \dots, \omega_{v,1z}, \omega_{v,21}, \omega_{v,23}, \dots, \omega_{v,2z}, \dots, \omega_{v,zz-1}$, are randomly chosen from the mating pool to yield their offspring. Based on a crossover rate, new sigmoid slopes and interconnection weights are generated as follows:

$$\omega_u = \gamma \omega_u + (1 - \gamma) \omega_v, \omega_v = (1 - \gamma) \omega_u + \gamma \omega_v \quad (28)$$

$$w_{u,ij} = \varphi_{ij} w_{u,ij} + (1 - \varphi_{ij}) w_{v,ij}, w_{v,ij} = (1 - \varphi_{ij}) w_{u,ij} + \varphi_{ij} w_{v,ij} \quad (29)$$

in which γ, φ_{ij} , and h_k ($1 \leq i, j, k \leq z$) are values randomly selected from $[0, 1]$. With the pre-specified mutation rate, each parameter is altered by adding or subtracting a small real value. Likewise, zero is assigned to α , and we perform Steps 5–9 for each newly generated chromosome.

4. Empirical Results

The purpose of an empirical setting is to substantiate whether the proposed weighting scheme (i.e., GF) is instrumental for improving the forecasting accuracy of decomposition ensemble models with linear addition. For this, monthly data on air passenger flows from Taiwan's six primary airports—Taoyuan, Kaohsiung, Songshan, Taichung, Kinmen, and Penghu—from January 2014 to December 2019 were obtained from Taiwan civil air transportation statistics. The sample period ceased in December 2019 to avoid the massive impact of outliers arising due to the COVID-19 pandemic. The dataset was partitioned into in-sample data that covered the periods from January 2014 to April 2018, constituting 72.2% of the original data, and out-of-sample data from May 2018 to December 2019. All data were measured in units of ten thousand passengers.

We compared the accuracy of five primary FCM-based decomposition ensemble models: MLP-GF, SVR-GF, RBFN-GF, CNN-GF, and LSTM-GF, referring to a decomposition ensemble model that applies an MLP, SVR, RBFN, CNN, and LSTM to component series forecasting, respectively, and that incorporates the proposed weighting scheme into linear addition to produce ensemble forecasts. To examine the forecasting performance for weighting schemes, other useful correlation analyses were taken into account: namely, equal weights, the Pearson correlation, and the Kendall correlation. In addition to the MLP, four decomposition ensemble models were treated as benchmark models for MLP-GF: namely, MLP-E, MLP-G, MLP-P, and MLP-K, referring to an MLP that incorporates equal weights, GRA, the Pearson correlation, and Kendall correlation, respectively, into linear addition to produce ensemble forecasts. Likewise, SVR-GF, RBFN-GF, CNN-GF, and LSTM have their own competitors.

In terms of the RMSE, Table 2 shows the forecasting accuracy of single neural networks. Tables 3–6 summarize the accuracy of different forecasting models for six airports and across four forecasting horizons (one, three, six, and 12 months ahead). Because the Friedman test is a robust non-parametric test suitable for comparing multiple classifiers [51], it was used to validate whether or not the results of the MLP-based models have a

significant difference. The Friedman test ranked a forecasting method for each time series separately, the best-performing method had the rank of 1, the second best had a rank of 2, and so forth. Table 7 reveals that the Friedman statistic is greater than the critical value of $F(5, 115)$ at a 1% level of significance, so that the forecasting results of the six MLP-based models have a significant difference at a 99% confidence level. Tables 8–11 respectively show that the forecasting results of the SVR-, RBFN-, CNN-, and LSTM-based models were significantly different at a 99% confidence level.

Table 2. Forecasting accuracy of different NNs.

Horizon	Models	Taoyuan	Kaohsiung	Songshan	Taichung	Kinmen	Penghu
1-step	MLP	17.384	4.637	4.096	2.783	2.914	2.559
	SVR	34.582	6.390	5.343	3.441	2.872	4.691
	RBFN	16.151	4.229	3.590	2.782	2.233	2.463
	CNN	38.310	8.909	4.167	3.502	2.373	3.877
	LSTM	38.961	19.663	4.691	3.653	3.001	3.378
3-step	MLP	11.432	7.866	8.148	3.872	3.448	2.912
	SVR	33.080	6.862	6.079	3.943	2.938	5.013
	RBFN	12.456	4.508	5.875	3.926	2.620	2.599
	CNN	37.157	9.025	4.807	3.776	2.358	4.180
	LSTM	40.405	8.433	6.025	4.209	2.652	3.959
6-step	MLP	9.822	5.507	6.462	3.149	2.243	3.052
	SVR	31.314	7.176	5.828	3.607	2.365	4.896
	RBFN	16.566	5.889	5.953	2.803	2.066	3.332
	CNN	41.632	8.681	5.412	3.938	2.438	3.499
	LSTM	44.103	7.257	6.227	4.183	2.559	4.147
12-step	MLP	12.293	65.852	3.879	3.947	2.783	2.351
	SVR	28.111	7.799	5.248	3.619	2.195	4.714
	RBFN	11.590	6.241	2.624	2.210	1.771	2.558
	CNN	94.517	8.303	2.632	3.745	1.903	6.989
	LSTM	97.319	8.977	2.573	8.235	1.472	3.696

Table 3. Accuracy of decomposition ensemble models for one-step-ahead forecasting.

Models	Taoyuan	Kaohsiung	Songshan	Taichung	Kinmen	Penghu
MLP-E	14.356	3.181	21.299	1.058	1.945	3.006
MLP-G	12.610	2.199	20.341	0.721	1.693	2.375
MLP-P	15.295	2.416	18.299	2.272	1.796	5.877
MLP-K	15.325	3.585	19.499	8.187	1.896	5.495
MLP-GF	11.784	2.117	19.754	0.597	1.570	1.498
SVR-E	23.321	2.200	4.115	0.822	1.859	3.043
SVR-G	19.082	2.460	4.278	1.042	1.838	2.770
SVR-P	20.958	2.829	5.072	2.099	1.947	6.094
SVR-K	21.946	4.025	5.249	8.753	1.968	5.648
SVR-GF	19.513	2.195	4.077	0.762	1.811	1.806
RBFN-E	10.843	2.955	2.553	0.505	1.410	3.578
RBFN-G	12.567	2.238	3.124	0.680	1.390	2.334
RBFN-P	14.175	2.480	4.532	2.195	1.742	5.669
RBFN-K	14.446	3.259	4.783	6.796	1.711	6.087
RBFN-GF	8.926	1.824	2.342	0.420	1.348	1.643
CNN-E	9.389	2.425	3.797	0.651	2.234	3.493
CNN-G	11.257	2.374	4.173	0.869	1.982	2.315
CNN-P	15.346	2.674	4.930	2.097	2.060	5.794

CNN-K	15.582	3.891	5.193	15.871	2.038	5.597
CNN-GF	9.550	2.263	3.808	0.649	1.806	1.985
LSTM-E	13.680	2.588	6.496	0.766	1.975	2.850
LSTM-G	11.598	1.874	5.497	1.092	1.889	2.226
LSTM-P	15.934	2.292	5.665	2.283	2.086	6.075
LSTM-K	16.451	3.447	5.766	18.165	2.295	4.978
LSTM-GF	10.481	1.710	5.443	0.692	1.737	1.682

Note: Bold values denote the most accurate models among the proposed model and its counterpart.

Table 4. Accuracy of decomposition ensemble models for three-step-ahead forecasting.

Models	Taoyuan	Kaohsiung	Songshan	Taichung	Kinmen	Penghu
MLP-E	11.968	2.561	4.478	1.050	2.036	4.314
MLP-G	13.669	2.552	4.857	0.928	1.797	2.894
MLP-P	15.692	2.998	5.151	2.257	1.867	6.286
MLP-K	15.768	3.937	5.688	2.184	1.890	6.279
MLP-GF	11.560	2.501	4.511	0.694	1.790	2.811
SVR-E	21.491	2.650	4.344	0.945	1.886	2.917
SVR-G	17.999	2.648	4.343	1.092	1.968	2.264
SVR-P	17.740	3.321	5.082	2.178	2.013	6.377
SVR-K	21.986	3.972	5.346	1.032	2.042	5.870
SVR-GF	18.427	2.639	4.120	0.963	1.892	1.797
RBFN-E	14.317	3.127	5.616	1.228	2.026	2.959
RBFN-G	14.062	2.300	5.335	1.195	1.847	2.594
RBFN-P	15.966	2.880	5.395	2.199	1.912	5.575
RBFN-K	16.474	4.070	5.550	2.949	1.922	5.446
RBFN-GF	11.962	2.213	5.292	1.131	1.797	1.838
CNN-E	17.385	2.941	3.954	0.962	2.341	3.292
CNN-G	15.843	2.912	4.191	1.049	2.155	2.213
CNN-P	16.513	3.407	4.899	1.990	2.286	6.463
CNN-K	16.775	3.642	5.272	12.714	2.344	6.093
CNN-GF	15.497	2.888	3.991	0.936	2.040	2.084
LSTM-E	16.789	3.116	5.290	1.851	1.904	2.561
LSTM-G	16.405	2.863	5.085	1.248	1.868	2.066
LSTM-P	16.909	3.390	5.185	2.570	1.869	6.027
LSTM-K	16.393	3.686	5.189	12.971	1.941	5.377
LSTM-GF	15.755	2.814	5.047	0.997	1.791	1.615

Note: Bold values denote the most accurate models among the proposed model and its counterpart.

Table 5. Accuracy of decomposition ensemble models for six-step-ahead forecasting.

Models	Taoyuan	Kaohsiung	Songshan	Taichung	Kinmen	Penghu
MLP-E	19.477	4.965	6.339	1.473	5.334	4.285
MLP-G	17.929	3.486	5.019	1.202	2.388	2.560
MLP-P	17.438	3.597	4.925	2.197	2.970	6.039
MLP-K	18.513	6.980	5.630	1.671	3.288	4.890
MLP-GF	16.994	3.778	4.524	1.198	3.096	2.476
SVR-E	20.081	3.315	5.138	1.458	2.287	2.962
SVR-G	18.789	3.242	4.098	1.160	2.194	3.236
SVR-P	17.819	3.898	5.127	2.199	2.357	6.396
SVR-K	34.865	5.540	5.469	12.461	2.427	5.093
SVR-GF	20.952	2.979	3.895	1.068	2.135	1.783
RBFN-E	15.144	3.797	14.144	1.554	2.639	2.878

RBFN-G	14.678	2.474	10.811	1.486	2.094	2.197
RBFN-P	16.939	3.152	7.355	2.178	2.091	6.442
RBFN-K	25.332	6.953	8.522	1.992	2.075	4.231
RBFN-GF	13.557	2.403	8.909	1.443	2.048	1.677
CNN-E	16.560	3.698	5.724	1.311	2.724	2.417
CNN-G	15.973	3.397	4.856	1.054	2.260	2.017
CNN-P	16.805	3.746	5.260	2.149	4.785	6.195
CNN-K	24.193	4.485	5.375	14.083	2.158	4.960
CNN-GF	15.146	3.340	4.815	1.025	2.194	1.592
LSTM-E	17.041	4.456	4.437	4.299	2.751	2.058
LSTM-G	16.519	3.785	3.735	2.383	2.270	1.979
LSTM-P	17.806	4.187	4.668	3.913	4.639	5.961
LSTM-K	27.295	5.398	5.098	14.826	3.071	4.696
LSTM-GF	14.976	3.583	3.525	2.808	2.240	1.701

Note: Bold values denote the most accurate models among the proposed model and its counterpart.

Table 6. Accuracy of decomposition ensemble models for 12-step-ahead forecasting.

Models	Taoyuan	Kaohsiung	Songshan	Taichung	Kinmen	Penghu
MLP-E	13.461	8.368	5.120	2.290	2.356	2.717
MLP-G	13.946	4.971	3.356	1.955	2.178	1.878
MLP-P	16.676	4.672	4.395	2.283	2.339	5.764
MLP-K	75.810	16.980	9.480	2.059	4.180	4.755
MLP-GF	12.274	4.692	3.203	1.898	2.143	1.821
SVR-E	18.081	3.294	2.142	5.500	3.734	4.276
SVR-G	17.774	3.063	2.692	5.042	3.364	3.781
SVR-P	17.596	3.696	4.542	5.752	3.423	6.617
SVR-K	63.232	16.494	13.359	7.199	2.812	9.482
SVR-GF	17.279	3.028	1.895	4.387	3.042	2.127
RBFN-E	16.414	8.319	17.791	1.846	3.575	2.807
RBFN-G	16.557	6.364	7.568	1.517	2.566	2.054
RBFN-P	17.326	5.273	7.801	2.254	2.795	6.607
RBFN-K	82.592	17.771	7.692	1.899	3.297	4.386
RBFN-GF	14.340	6.151	9.150	1.487	2.465	1.797
CNN-E	9.696	5.032	2.099	2.324	2.073	2.210
CNN-G	12.830	3.447	2.599	1.567	1.511	2.192
CNN-P	16.573	3.669	4.321	3.083	2.172	5.895
CNN-K	86.723	17.601	4.599	8.898	1.894	7.631
CNN-GF	9.888	3.407	2.064	1.498	1.532	1.691
LSTM-E	16.100	5.253	2.369	2.297	2.438	1.935
LSTM-G	14.558	4.342	2.655	1.558	2.074	2.734
LSTM-P	17.993	4.213	3.983	3.398	2.185	5.986
LSTM-K	90.259	18.613	7.293	9.292	3.343	3.983
LSTM-GF	12.842	3.993	2.286	1.394	1.933	1.374

Note: Bold values denote the most accurate models among the proposed model and its counterpart.

Table 7. Friedman test for MLP-based forecasting methods with average ranks.

Model	Friedman Statistic				
MLP	MLP-E	MLP-G	MLP-P	MLP-K	MLP-GF
4.25	4.042	2.458	3.875	4.875	1.5
19.245 ***					

Note: *** denotes that the forecasting results of these models have a significant difference at a 1% level of significance.

Table 8. Friedman test for SVR-based forecasting methods with average ranks.

Model						Friedman Statistic
SVR	SVR-E	SVR-G	SVR-P	SVR-K	SVR-GF	
5	3	2.458	4	5.083	1.458	34.586 ***

Note: *** denotes that the forecasting results of these models have a significant difference at a 1% level of significance.

Table 9. Friedman test for RBFN-based forecasting methods with average ranks.

Model						Friedman Statistic
RBFN	RBFN-E	RBFN-G	RBFN-P	RBFN-K	RBFN-GF	
4.042	3.875	2.625	4.208	4.875	1.417	20.391 ***

Note: *** denotes that the forecasting results of these models have a significant difference at a 1% level of significance.

Table 10. Friedman test for CNN-based forecasting methods with average ranks.

Model						Friedman Statistic
CNN	CNN-E	CNN-G	CNN-P	CNN-K	CNN-GF	
5	3.125	2.333	4.417	4.958	1.292	57.028 ***

Note: *** denotes that the forecasting results of these models have a significant difference at a 1% level of significance.

Table 11. Friedman test for LSTM-based forecasting methods.

Average Rank						Friedman Statistic
LSTM	LSTM-E	LSTM-G	LSTM-P	LSTM-K	LSTM-GF	
4.833	3.458	2.333	4.208	5.042	1.125	45.918 ***

Note: *** denotes that the forecasting results of these models have a significant difference at a 1% level of significance.

The Nemenyi test was further conducted to detect any significant differences between MLP-based models. The results of two forecasting methods are significantly different if their average ranks differ by at least the critical difference. Table 12 shows the difference in the average ranks between a matched pair of MLP-based models. For instance, MLP-GF significantly outperformed MLP-E because the average ranks of MLP-GF and MLP-E differ by 2.542, which is greater than the critical difference of 1.817 at a 99% confidence level. MLP-GF significantly outperformed MLP, MLP-E, MLP-P, and MLP-K as well. Tables 13–16 indicate the significant superiority of SVR-GF, RBFN-GF, CNN-GF, and LSTM-GF over their respective competitors. Overall, it is reasonable to conclude that the decomposition ensemble methods with the proposed weighting scheme outperformed the benchmark models in terms of forecasting air passenger flow in Taiwan. The Friedman test results presented in Tables 17–21 further demonstrate significant differences in forecasting performance across different horizons for the MLP-, SVR-, RBFN-, CNN-, and LSTM-based models, respectively. The subsequent Nemenyi post hoc test further confirms that the proposed FCM-based decomposition ensemble models outperform the benchmark models considered in this study.

Table 12. Results of the Nemenyi test for MLP-based models.

Method	MLP	MLP-E	MLP-G	MLP-P	MLP-K	MLP-GF
MLP	0	0.208	1.792 **	0.375	−0.625	2.750 ***
MLP-E	−0.208	0	1.583 *	0.167	−0.833	2.542 ***

MLP-G	-1.792	-1.583	0	-1.417	-2.417	0.958
MLP-P	-0.375	-0.167	1.417 *	0	-1.000	2.375 ***
MLP-K	0.625	0.833	2.417 ***	1.000	0	3.375 ***
MLP-GF	-2.750	-2.542	-0.958	-2.375	-3.375	0

Note: ***, **, and * denote that the performance of two methods is significantly different at 1%, 5%, and 10% levels if the corresponding average ranks differ at least the critical differences of 1.817, 1.539, and 1.398, respectively.

Table 13. Results of the Nemenyi test for SVR-based models.

Method	SVR	SVR-E	SVR-G	SVR-P	SVR-K	SVR-GF
SVR	0	2.000 ***	2.542 ***	1.000	-0.083	3.542 ***
SVR-E	-2.000	0	0.542	-1.000	-2.083	1.542 **
SVR-G	-2.542	-0.542	0	-1.542	-2.625	1.000
SVR-P	-1.000	1.000	1.542 **	0	-1.083	2.542 ***
SVR-K	0.083	2.083 ***	2.625 ***	1.083	0	3.625 ***
SVR-GF	-3.542	-1.542	-1.000	-2.542	-3.625	0

Note: *** and ** denote that the performance of two methods is significantly different at 1% and 5% levels if the corresponding average ranks differ at least the critical differences of 1.817 and 1.539, respectively.

Table 14. Results of the Nemenyi test for RBFN-based models.

Method	RBFN	RBFN-E	RBFN-G	RBFN-P	RBFN-K	RBFN-GF
RBFN	0	0.167	1.417 *	-0.167	-0.833	2.625 ***
RBFN-E	-0.167	0	1.250	-0.333	-1.000	2.458 ***
RBFN-G	-1.417	-1.250	0	-1.583	-2.250	1.208
RBFN-P	0.167	0.333	1.583 **	0.	-0.667	2.792 ***
RBFN-K	0.833	1.000	2.250 ***	0.667	0	3.458 ***
RBFN-GF	-2.625	-2.458	-1.208	-2.792	-3.458	0

Note: ***, **, and * denote that the performance of two methods is significantly different at 1%, 5%, and 10% levels if the corresponding average ranks differ at least the critical differences of 1.817, 1.539, and 1.398, respectively.

Table 15. Results of the Nemenyi test for CNN-based models.

Method	CNN	CNN-E	CNN-G	CNN-P	CNN-K	CNN-GF
CNN	0	1.875 ***	2.667 ***	0.583	0.042	3.708 ***
CNN-E	-1.875	0	0.792	-1.292	-1.833	1.833 ***
CNN-G	-2.667	-0.792	0	-2.083	-2.625	1.042
CNN-P	-0.583	1.292	2.083 ***	0	-0.542	3.125 ***
CNN-K	-0.042	1.833 ***	2.625 ***	0.542	0	3.667 ***
CNN-GF	-3.708	-1.833	-1.042	-3.125	-3.667	0

Note: *** denotes that the performance of two methods is significantly different at 1% level if the corresponding average ranks differ at least the critical differences of 1.817.

Table 16. Results of the Nemenyi test for LSTM-based models.

Method	LSTM	LSTM-E	LSTM-G	LSTM-P	LSTM-K	LSTM-GF
LSTM	0	1.375	2.500 ***	0.625	-0.208	3.708 ***
LSTM-E	-1.375	0	1.125	-0.750	-1.583	2.333 ***
LSTM-G	-2.500	-1.125	0	-1.875	-2.708	1.208
LSTM-P	-0.625	0.750	1.875 ***	0	-0.833	3.083 ***
LSTM-K	0.208	1.583 **	2.708 ***	0.833	0	3.917 ***

LSTM-GF	−3.708	−2.333	−1.208	−3.083	−3.917	0
---------	--------	--------	--------	--------	--------	---

Note: *** and ** denote that the performance of two methods is significantly different at 1% and 5% levels if the corresponding average ranks differ at least the critical differences of 1.817 and 1.539, respectively.

The null hypothesis for the Diebold–Mariano test, which stated that the proposed FCM-based model did not outperform any of its corresponding forecasting models in accuracy, was tested. Let MLP-GF, SVR-GF, RBFN-GF, CNN-GF, and LSTM-GF represent the proposed FCM-based models for the MLP-, SVR-, RBFN-, CNN-, and LSTM-based models, respectively. Thus, for instance, the competitors of the MLP-GF include the MLP, MLP-E, MLP-G, MLP-P, and MLP-K. As Tables 22–25 illustrate, the Diebold–Mariano test statistics were positive for the majority of cases, which suggested that the proposed FCM-based models were more accurate than their competitors. The ratios of statistically positive statistics were 77.33%, 60.67%, 68%, and 72% in the cases of the one-, three-, six-, and 12-step-ahead forecasts, respectively. These results support the superiority of the proposed FCM-based models in terms of accuracy over the other forecasting models considered across all horizons.

Table 17. Friedman test for MLP-based forecasting methods with average ranks across different horizons.

Horizon	Model						Friedman Statistic
	MLP	MLP-E	MLP-G	MLP-P	MLP-K	MLP-GF	
1-step	4.5	4.167	2.5	3.667	4.667	1.5	4.052 ***
3-step	4.667	3.167	2.5	4.5	4.833	1.333	6.581 ***
6-step	3.667	4.667	2.333	3.5	4.833	2	3.161 **
12-step	4.167	4.167	2.5	3.833	5.167	1.167	7.023 ***

Note: *** and ** denote that the performance of six methods is significantly different at 1% and 5% levels, respectively.

Table 18. Friedman test for SVR-based forecasting methods with average ranks across different horizons.

Horizon	Model						Friedman Statistic
	SVR	SVR-E	SVR-G	SVR-P	SVR-K	SVR-GF	
1-step	5	2.833	2.333	4.167	5	1.167	4.937 ***
3-step	5.667	2.5	2.5	4	4.667	1.667	10 ***
6-step	5.167	3	2.167	3.667	5.5	1.5	13.75 ***
12-step	3.667	3.667	2.833	4.167	5.167	1.5	3.949 ***

Note: *** denotes that the performance of six methods is significantly different at 1% level.

Table 19. Friedman test for RBFN-based forecasting methods with average ranks across different horizons.

Horizon	Model						Friedman Statistic
	RBFN	RBFN-E	RBFN-G	RBFN-P	RBFN-K	RBFN-GF	
1-step	5	2.833	2.5	4.333	5.333	1	20 ***
3-step	5	4	2.167	4	4.833	1	12.697 ***
6-step	3.833	4.167	3	4.167	4.5	1.5	3.678 **
12-step	2.333	4.5	2.833	4.333	4.833	2.167	3.378 **

Note: *** and ** denote that the performance of six methods is significantly different at 1% and 5% levels, respectively.

Table 20. Friedman test for CNN-based forecasting methods with average ranks across different horizons.

Horizon	Model						Friedman Statistic
	CNN	CNN-E	CNN-G	CNN-P	CNN-K	CNN-GF	
1-step	5	2.667	2.833	4.667	5	1.333	32.952 ***
3-step	5.167	3	2.333	4.167	5.167	1.167	14.688 ***
6-step	5	3.833	2	4.5	4.333	1.333	8.462 ***
12-step	5	3	2.167	4.333	5.333	1.333	21.471 ***

Note: *** denotes that the performance of six methods is significantly different at 1% level.

Table 21. Friedman test for LSTM-based forecasting methods across different horizons.

Horizon	Models						Friedman Statistic
	LSTM	LSTM-E	LSTM-G	LSTM-P	LSTM-K	LSTM-GF	
1-step	4.667	3.5	2.333	4.167	5.167	1.167	9.318 ***
3-step	5.667	3.5	2.333	4.167	4.333	1	16.575 ***
6-step	4.833	3.667	1.833	4.333	5.167	1.167	16.875 ***
12-step	4.167	3.167	2.833	4.167	5.5	1.167	8.235 ***

Note: *** denotes that the performance of six methods is significantly different at 1% level.

Table 22. Results of Diebold–Mariano test statistic for one-step-ahead forecasting.

Models	Taoyuan	Kaohsiung	Songshan	Taichung	Kinmen	Penghu
MLP-based						
MLP	1.510 *	3.557 ***	−3.230	5.280 ***	3.174 ***	2.152 **
MLP-E	1.202	1.826 **	4.737 ***	3.547 ***	1.650 **	2.540 ***
MLP-G	0.886	1.094	2.239 **	1.543 *	0.826	2.538 ***
MLP-P	1.532 *	1.252	−2.315	3.792 ***	1.544 *	5.275 ***
MLP-K	1.705 **	2.959 ***	−0.553	11.193 ***	1.589 *	5.573 ***
SVR-based						
SVR	5.139 ***	4.880 ***	1.460 *	7.441 ***	2.830 ***	3.661 ***
SVR-E	2.787 ***	0.047	0.123	1.194	0.519	1.958 **
SVR-G	−0.424	0.896	2.173 **	2.316 ***	0.646	2.680 ***
SVR-P	1.587 *	1.831 **	4.142 ***	4.071 ***	1.531 *	4.668 ***
SVR-K	2.863 ***	3.958 ***	4.539 ***	9.428 ***	1.155	5.012 ***
RBFN-based						
RBFN	2.480 ***	3.889 ***	1.992 **	3.237 ***	3.301 ***	1.921 **
RBFN-E	2.180 **	2.031 **	2.352 ***	1.992 **	0.456	2.745 ***
RBFN-G	1.893 **	1.167	2.394 ***	3.580 ***	0.580	1.808 **
RBFN-P	1.766 **	1.608 *	3.849 ***	4.035 ***	2.451 ***	5.279 ***
RBFN-K	1.989 **	2.324 **	3.725 ***	9.033 ***	2.130 **	5.052 ***
CNN-based						
CNN	5.508 ***	4.976 ***	0.532	6.757 ***	1.651 **	4.117 ***
CNN-E	−0.144	0.688	−0.211	0.097	2.357 ***	3.124 ***
CNN-G	2.581 ***	0.709	2.194 **	2.481 ***	2.020 **	0.906
CNN-P	2.785 ***	1.273	3.475 ***	3.834 ***	1.726 **	4.850 ***
CNN-K	2.806 ***	3.075 ***	3.491 ***	17.261 ***	1.232	4.753 ***
LSTM-based						
LSTM	5.654 ***	10.488 ***	−0.916	6.429 ***	3.146 ***	2.955 ***
LSTM-E	2.261 **	1.486 **	1.368 *	0.698	1.747 **	2.003 **
LSTM-G	1.215	2.388 ***	0.314	2.340 ***	1.250	1.544 **
LSTM-P	1.589 *	1.953 **	0.642	3.906 ***	1.691 **	5.037 ***
LSTM-K	1.484 *	2.981 ***	0.863	20.172 ***	2.052 **	4.754 ***

Note: A negative sign of the statistic implies that the proposed FCM-based models performed worse. ***, **, and * denote significance at 1%, 5%, and 10% levels, respectively.

Table 23. Results of Diebold–Mariano test statistic for three-step-ahead forecasting.

Models	Taoyuan	Kaohsiung	Songshan	Taichung	Kinmen	Penghu
MLP-based						
MLP	−0.063	4.469 ***	2.517 ***	5.871 ***	3.080 ***	0.239
MLP-E	0.648	0.343	−0.303	1.973 **	1.119	2.931 ***
MLP-G	1.749 **	0.348	0.979	2.065 **	0.237	0.426
MLP-P	1.789 **	1.586 *	1.619 *	3.798 ***	1.153	4.176 ***
MLP-K	1.987 **	2.828 ***	2.118 **	3.608 ***	0.989	4.074 ***
SVR-based						
SVR	4.606 ***	4.453 ***	1.707 **	8.641 ***	2.321 ***	4.017 ***
SVR-E	1.945 **	0.165	1.187	−0.275	−0.111	2.591 ***
SVR-G	−0.944	0.121	1.707 **	1.553 *	0.505	2.563 ***
SVR-P	−0.594	1.719 **	2.430 ***	3.770 ***	0.604	4.785 ***
SVR-K	3.082 ***	2.647 ***	2.474 ***	10.863 ***	0.731	4.838 ***
RBFN-based						
RBFN	0.240	3.189 ***	0.588	8.641 ***	2.423 ***	4.017 ***
RBFN-E	2.187 **	2.089 **	0.681	−0.275	1.445 *	2.591 ***
RBFN-G	1.155	0.570	1.155	1.553 *	0.849	2.563 **
RBFN-P	1.644 **	1.776 **	0.512	3.770 ***	1.076	4.785 ***
RBFN-K	1.945 **	3.079 ***	0.974	10.863 ***	1.202	4.838 ***
CNN-based						
CNN	4.743 ***	6.947 ***	1.137	6.668 ***	1.230	4.260 ***
CNN-E	1.423 *	0.354	−0.431	1.105	1.779 **	2.770 ***
CNN-G	0.422	0.456	1.525 *	1.280 *	0.840	0.728
CNN-P	0.723	1.680 **	2.044 **	3.329 ***	1.514 *	4.361 ***
CNN-K	0.695	2.195 **	2.599 ***	13.662 ***	1.612 *	4.527 ***
LSTM-based						
LSTM	5.052 ***	5.876 ***	1.288 *	6.582 ***	2.602 ***	3.566 ***
LSTM-E	0.826	0.741	0.793	3.748 ***	0.885	2.428 ***
LSTM-G	0.693	0.303	0.472	2.786 ***	0.511	1.739 **
LSTM-P	1.045	1.425 *	0.409	4.274 ***	0.577	5.212 ***
LSTM-K	0.808	2.023 **	0.719	13.872 ***	0.866	4.955 ***

Note: A negative sign of the statistic implies that the proposed FCM-based models performed worse. ***, **, and * denote significance at 1%, 5%, and 10% levels, respectively.

Table 24. Results of Diebold–Mariano test statistic for six-step-ahead forecasting.

Models	Taoyuan	Kaohsiung	Songshan	Taichung	Kinmen	Penghu
MLP-based						
MLP	−2.215	2.432 ***	1.565 *	8.951 ***	−1.328	1.220
MLP-E	1.297 *	2.163 **	2.579 ***	2.006 **	2.004 **	2.164 **
MLP-G	0.715	−0.684	1.669 **	0.035	−2.156	1.656 **
MLP-P	0.324	−0.600	1.287 *	3.240 ***	−0.851	4.644 ***
MLP-K	0.906	3.683 ***	2.073 **	2.239 ***	0.786	3.993 ***
SVR-based						
SVR	4.841 ***	4.212 ***	2.086 **	6.558 ***	0.466	4.419 ***
SVR-E	−2.427	1.374 *	2.142 **	3.171 ***	1.062	2.552 ***
SVR-G	−1.578	1.513 *	1.674 **	1.028	1.155	2.720 ***
SVR-P	−1.709	1.851 **	2.980 ***	3.227 ***	1.667 **	4.540 ***

SVR-K	5.668 ***	3.036 ***	3.082 ***	13.717 ***	1.839 **	4.266 ***
RBFN-based						
RBFN	1.050	4.670 ***	−1.901	4.258 ***	0.049	3.154 ***
RBFN-E	1.890 **	2.097 **	2.815 ***	1.361 *	2.361 ***	2.847 ***
RBFN-G	0.531	0.815	2.622 ***	0.445	0.772	1.692 **
RBFN-P	1.163	2.152 **	−1.930	2.109 **	0.936	4.639 ***
RBFN-K	3.933 ***	4.496 ***	−0.900	2.034 **	0.993	4.085 ***
CNN-based						
CNN	4.915 ***	5.924 ***	0.787	6.764 ***	0.865	5.001 ***
CNN-E	0.926	1.054	1.433 *	2.089 **	2.956 ***	2.808 ***
CNN-G	1.171	0.638	0.182	1.146	0.746	1.524 *
CNN-P	1.152	1.337 *	0.837	3.818 ***	6.225 ***	4.928 ***
CNN-K	3.119 ***	2.127 **	0.963	14.789 ***	−0.342	4.614 ***
LSTM-based						
LSTM	6.245 ***	3.646 ***	3.713 ***	3.396 ***	1.362 *	4.142 ***
LSTM-E	2.264 **	1.344 *	1.397 *	7.152 ***	2.048 **	1.747 **
LSTM-G	1.613 *	1.242	1.241	−2.763	2.094 **	0.911
LSTM-P	1.430 *	1.319 *	2.799 ***	3.177 ***	6.066 ***	4.742 ***
LSTM-K	4.124 ***	2.233 **	3.079 ***	13.734 ***	3.955 ***	4.292 ***

Note: A negative sign of the statistic implies that the proposed FCM-based models performed worse. ***, **, and * denote significance at 1%, 5%, and 10% levels, respectively.

Table 25. Results of Diebold–Mariano test statistic for 12-step-ahead forecasting.

Models	Taoyuan	Kaohsiung	Songshan	Taichung	Kinmen	Penghu
MLP-based						
MLP	0.007	1.068	0.713	3.334 ***	1.507 *	1.135
MLP-E	0.696	2.326 ***	2.995 **	1.553 *	1.256	2.457 ***
MLP-G	1.958 **	1.677 **	3.047 ***	4.128 ***	0.618	0.223
MLP-P	2.966 ***	−0.113	3.654 ***	1.618 *	1.260	4.631 ***
MLP-K	10.465 ***	8.874 ***	4.241 ***	1.142	3.491 ***	3.990 ***
SVR-based						
SVR	4.192 ***	4.605 ***	3.446 ***	−1.791	−1.119	4.619 ***
SVR-E	2.175 **	0.886	0.981	5.703 ***	3.891 ***	3.543 ***
SVR-G	0.177	0.478	2.782 ***	2.673 ***	3.028 ***	3.149 ***
SVR-P	0.313	2.040 **	4.727 ***	3.080 ***	2.923 ***	4.463 ***
SVR-K	9.210 ***	10.194 ***	6.668 ***	6.729 ***	−1.425	5.022 ***
RBFN-based						
RBFN	−1.270	0.075	−2.606	2.322 ***	−1.258	2.118 **
RBFN-E	1.344 *	2.084 **	3.260 ***	1.298 *	4.024 ***	3.144 ***
RBFN-G	2.015 **	0.367	−2.171	0.330	0.553	0.704
RBFN-P	1.371 *	−0.857	−2.214	2.881 ***	1.907 **	4.349 ***
RBFN-K	10.251 ***	8.360	−0.906	2.039 **	1.295 *	3.359 ***
CNN-based						
CNN	12.046 ***	4.824 ***	1.170	6.483 ***	1.803 **	3.916 ***
CNN-E	−0.234	2.308 ***	0.295	3.461 ***	3.313 ***	2.166 **
CNN-G	1.831 **	0.134	1.907 **	3.240 ***	−0.219	1.973 **
CNN-P	2.498 ***	0.837	3.997 ***	4.037 ***	2.277 **	4.794 ***
CNN-K	12.277 ***	10.606 ***	3.982 ***	10.873 ***	1.617 *	5.598 ***
LSTM-based						
LSTM	12.331 ***	4.069 ***	0.951	9.065 ***	−1.806	3.035 ***
LSTM-E	1.543 *	2.061 **	0.820	3.800 ***	2.822 ***	1.472 *
LSTM-G	1.358 *	2.020 **	1.976 **	2.617 ***	1.313 *	2.943 ***

LSTM-P	2.929 ***	0.698	3.338 ***	4.181 ***	1.813 **	5.604 ***
LSTM-K	12.050 ***	10.111 ***	4.212 ***	10.992 ***	4.079 ***	4.464 ***

Note: A negative sign of the statistic implies that the proposed FCM-based models performed worse. ***, **, and * denote significance at 1%, 5%, and 10% levels, respectively.

5. Concluding Remarks

A well-performing forecasting model is beneficial to the airline industry for formulating appropriate developmental strategies and making operational plans. Decomposition ensemble models have been used to improve the accuracy of forecasts of air passenger demand. Such models in previous studies commonly use linear addition and simply add the individual single-component forecasts together. Our study investigated whether incorporating correlation analysis can enhance linear addition to produce competitive results for decomposition and ensemble modeling. Recently, Hu [50] applied GRA to gauge the correlation between forecasted component series and the original series to improve the forecasting accuracy of decomposition ensemble models. The novelty of our research was to develop a new weighting scheme that uses FCMs for decomposition and ensemble modeling to determine the final weights of individual forecasts by taking GRGs as the initial state values and then identifying the interrelationships among component series models.

The results in Tables 12–16 elucidated that decomposition ensemble models using the proposed weighting scheme (e.g., CNN-GF) significantly outperformed their respective competitors using the other considered weighting schemes, including equal weights, the Pearson correlation, and the Kendall correlation, across different forecasting horizons (e.g., CNN-E, CNN-G, CNN-P, CNN-K). Interestingly, the forecasting models with the GRA-based weighting schemes (e.g., CNN-G and CNN-GF) significantly outperformed their respective competitors with Pearson's and Kendall's correlations (e.g., CNN-E, CNN-P, CNN-K) as well. These findings suggest that the GRA-based weighting scheme outperforms those based on Pearson's and Kendall's correlation coefficients. This advantage may arise because GRA can simultaneously evaluate the relational degrees between the original time series and multiple forecasted component series, whereas Pearson's and Kendall's correlation coefficients measure the pairwise association between the original time series and each forecasted component series separately. Moreover, although MLP-GF did not significantly outperform MLP-G, it achieved a better average rank (1.500 vs. 2.458). Similarly, SVR-GF, RBFN-GF, CNN-GF, and LSTM-GF achieved better average ranks than their respective SVR-G, RBFN-G, CNN-G, and LSTM-G counterparts. These findings suggest that determining ensemble weights by considering the causal relationships among component series forecasting models can improve the forecasting performance of decomposition-based ensemble models.

The findings of this study also provide promising directions for future research on decomposition ensemble learning with correlation measures:

- (1) Because explanatory factors are likely to affect the accuracy of forecasting models, it is feasible to consider the effects on air transport of explanatory factors such as economic growth.
- (2) AI-based models such as Elman neural networks [52], least-squares support vector machines [11], the non-additive Choquet integral [28], and grey forecasting [53–55] have been widely employed to produce forecasts for time series. Our future research will consider incorporating these forecasting models into component series forecasting to strengthen the overall accuracy of the proposed FCM-based models.
- (3) For the 12-step-ahead forecasting, MLP-K, SVR-K, RBFN-K, CNN-K, and LSTM-K produced a greater number of anomalous forecasts for Taoyuan Airport. One possible explanation is that these models, which employ Kendall correlation-based

weighting, may be more susceptible to overfitting. Future research could investigate whether decomposition-based ensemble models using Kendall correlation-based weighting are more likely to generate anomalous forecasts at longer forecasting horizons.

- (4) More recent forecasting models, including DLinear, N-HiTS, PatchTST, TimesNet, and iTransformer, represent advances in deep learning techniques. Future studies should compare decomposition-based ensemble models with these architectures using identical data partitions, forecasting horizons, and performance metrics.

Author Contributions: Conceptualization, Y.-C.H.; methodology, Y.-C.H.; software, G.W.; validation, Y.-C.H., G.W. and Y.-C.C.; formal analysis, Y.-C.H.; investigation, Y.-C.H. and G.W.; data curation, Y.-C.H.; writing—original draft preparation, Y.-C.H. and Y.-C.C.; writing—review and editing, Y.-C.H. and G.W.; visualization, G.W. and Y.-C.C.; supervision, Y.-C.H.; project administration, Y.-C.H.; funding acquisition, Y.-C.H. and G.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Science and Technology Council, Taiwan under grant NSTC 114-2410-H-033-021-MY2, and by the Zhejiang Provincial Statistical Youth Project under grant 24TJQN22.

Data Availability Statement: The data that support the findings of this study are openly available in the Taiwan Traffic of Civil Aviation by Airports at <https://www.caa.gov.tw/Article.aspx?a=1757&lang=2> (accessed on 1 September 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. He, H.; Chen, L.; Wang, S. Flight short-term booking demand forecasting based on a long short-term network. *Comput. Ind. Eng.* **2023**, *186*, 109707. <https://doi.org/10.1016/j.cie.2023.109707>.
2. Pereira, L.N.; Cerqueira, V. Forecasting hotel demand for revenue management using machine learning regression methods. *Curr. Issues Tour.* **2022**, *25*, 2733–2750. <https://doi.org/10.1080/13683500.2021.1999397>.
3. Jin, F.; Li, Y.; Sun, S.; Li, H. Forecasting air passenger demand with a new hybrid ensemble approach. *J. Air Transp. Manag.* **2020**, *83*, 101744. <https://doi.org/10.1016/j.jairtraman.2019.101744>.
4. Hu, Y.C. Air passenger flow forecasting using nonadditive forecast combination with grey prediction. *J. Air Transp. Manag.* **2023**, *112*, 102439. <https://doi.org/10.1016/j.jairtraman.2023.102439>.
5. Sun, S.; Lu, H.; Tsui, K.L.; Wang, S. Nonlinear vector auto-regression neural network for forecasting air passenger flow. *J. Air Transp. Manag.* **2019**, *78*, 54–62. <https://doi.org/10.1016/j.jairtraman.2019.04.005>.
6. Bi, J.W.; Han, T.Y.; Yao, Y.B. Fine-grained tourism demand forecasting: A decomposition ensemble deep learning model. *Tour. Econ.* **2023**, *29*, 1736–1763. <https://doi.org/10.1177/13548166231158705>.
7. Xie, G.; Liu, S.; Lin, X. An interval decomposition-ensemble model for tourism forecasting. *J. Hosp. Tour. Res.* **2023**, *49*, 600–614. <https://doi.org/10.1177/10963480231198539>.
8. Hu, Y.C.; Wu, G.; Tsai, J.F. Incorporating grey relational analysis into decomposition ensemble methods for forecasting air passenger demand. *Grey Syst. Theory Appl.* **2025**, *15*, 185–207. <https://doi.org/10.1108/GS-07-2024-0092>.
9. Tang, L.; Zhang, C.; Li, L.; Wang, S. A multi-scale method for forecasting oil price with multi-factor search engine data. *Appl. Energy* **2020**, *257*, 114033. <https://doi.org/10.1016/j.apenergy.2019.114033>.
10. Xing, G.; Sun, S.; Bi, D.; Guo, J.; Wang, S. Seasonal and trend forecasting of tourist arrivals: An adaptive multiscale ensemble learning approach. *Int. J. Tour. Res.* **2022**, *24*, 425–442. <https://doi.org/10.1002/jtr.2512>.
11. Liang, X.; Zhang, Q.; Hong, C.; Niu, W.; Yang, M. Do Internet search data help forecast air passenger demand? Evidence from China's airports. *Front. Psychol.* **2022**, *13*, 809954. <https://doi.org/10.3389/fpsyg.2022.809954>.
12. Li, X.; Zhang, X.; Zhang, C.; Wang, S. Forecasting tourism demand with a novel robust decomposition and ensemble framework. *Expert Syst. Appl.* **2024**, *236*, 121388. <https://doi.org/10.1016/j.eswa.2023.121388>.
13. Nazarieh, F.; Dehkordi, M.N. Time series forecasting based on a novel ensemble-based network and variational mode decomposition. *Expert Syst.* **2023**, *40*, e13291. <https://doi.org/10.1111/exsy.13291>.

14. Cao, Y.; Hou, X.; Chen, N. Short-term forecast of OD passenger flow based on ensemble empirical mode decomposition. *Sustainability* **2022**, *14*, 8562. <https://doi.org/10.3390/su14148562>.
15. Liu, J.; Tang, X.; Liu, H. Enhanced forecasting of online car-hailing demand using an improved empirical mode decomposition with long short-term memory neural network. *Transp. Lett.* **2024**, *16*, 1389–1405. <https://doi.org/10.1080/19427867.2024.2313832>.
16. Wei, Y.; Chen, M.C. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. *Transp. Res. Part C* **2012**, *21*, 148–162. <https://doi.org/10.1016/j.trc.2011.06.009>.
17. Yuan, Y.; Jiang, X.; Zhang, P.; Lai, C.S. Short-term high-speed rail passenger flow prediction by integrating ensemble empirical mode decomposition with multivariate grey support vector machine. *Eng. Appl. Artif. Intell.* **2024**, *136*, 109005. <https://doi.org/10.1016/j.engappai.2024.109005>.
18. Zhang, C.; Jiang, F.; Wang, S.; Sun, S. A new decomposition ensemble approach for tourism demand forecasting: Evidence from major source countries in Asia-Pacific region. *Int. J. Tour. Res.* **2021**, *23*, 832–845. <https://doi.org/10.1002/jtr.2445>.
19. Qin, L.; Li, W.; Li, S. Effective passenger flow forecasting using STL and ESN based on two improvement strategies. *Neurocomputing* **2019**, *356*, 244–256. <https://doi.org/10.1016/j.neucom.2019.04.061>.
20. Shi, Z.; Zhang, N.; Schonfeld, P.M.; Zhang, J. Short-term metro passenger flow forecasting using ensemble-chaos support vector regression. *Transp. A Transp. Sci.* **2020**, *16*, 194–212. <https://doi.org/10.1080/23249935.2019.1692956>.
21. Xie, G.; Wang, S.; Lai, K.K. Short-term forecasting of air passenger by using hybrid seasonal decomposition and least squares support vector regression approaches. *J. Air Transp. Manag.* **2014**, *37*, 20–26. <https://doi.org/10.1016/j.jairtraman.2014.01.009>.
22. Tang, Q.; Yang, M.; Yang, Y. ST-LSTM: A deep learning approach combined spatio-temporal features for short-term forecast in rail transit. *J. Adv. Transp.* **2019**, *2019*, 8392592. <https://doi.org/10.1155/2019/8392592>.
23. Wu, S.; Wang, W.; Song, Y.; Liu, S. An EEMD-LSTM, SVR, and BP decomposition ensemble model for steel future prices forecasting. *Expert Syst.* **2024**, *41*, e13672. <https://doi.org/10.1111/exsy.13672>.
24. Yang, D.; Chen, K.; Yang, M.; Zhao, X. Urban rail transit passenger flow forecast based on LSTM with enhanced long-term features. *IET Intell. Transp. Syst.* **2019**, *13*, 1475–1482. <https://doi.org/10.1049/iet-its.2018.5511>.
25. Li, H.; Wang, Y.; Xu, X.; Qin, L.; Zhang, H. Short-term passenger flow prediction under passenger flow control using a dynamic radial basis function network. *Appl. Soft Comput.* **2019**, *83*, 105620. <https://doi.org/10.1016/j.asoc.2019.105620>.
26. Hu, Y.C. Nonadditive tourism forecast combination using grey relational analysis. *Grey Syst. Theory Appl.* **2023**, *13*, 277–296. <https://doi.org/10.1108/GS-07-2022-0079>.
27. Hu, A.; Xie, N. Construction and application of a novel grey relational analysis model considering factor coupling relationship. *Grey Syst. Theory Appl.* **2024**, *15*, 1–20. <https://doi.org/10.1108/GS-04-2024-0046>.
28. Jiang, P.; Hu, Y.C.; Yen, G.F.; Jiang, H.; Chiu, Y.J. Using a novel Grey DANP model to identify interactions between manufacturing and logistics industries in China. *Sustainability* **2018**, *10*, 3456. <https://doi.org/10.3390/su10103456>.
29. Liu, S.; Yang, Y.; Forrest, J. *Grey Data Analysis: Methods, Models and Applications*; Springer: Berlin, Germany, 2017. <https://doi.org/10.1007/978-981-10-1841-1>.
30. Tzeng, G.H.; Shen, K.Y. *New Concepts and Trends of Hybrid Multiple Criteria Decision Making*; CRC Press: New York, NY, USA, 2017. <https://doi.org/10.1201/9781315166650>.
31. Kosko, B. Fuzzy cognitive maps. *Int. J. Man-Mach. Stud.* **1986**, *24*, 65–75. [https://doi.org/10.1016/S0020-7373\(86\)80040-2](https://doi.org/10.1016/S0020-7373(86)80040-2).
32. Bağdatlı, M.E.C.; Dokuz, A.Ş. Modeling discretionary lane-changing decisions using an improved fuzzy cognitive map with association rule mining. *Transp. Lett.* **2021**, *13*, 623–633. <https://doi.org/10.1080/19427867.2021.1919469>.
33. Salmeron, J.L.; Palos-Sanchez, P.R. Uncertainty propagation in fuzzy grey cognitive maps with Hebbian-like learning algorithms. *IEEE Trans. Cybern.* **2019**, *49*, 211–220. <https://doi.org/10.1109/TCYB.2017.2771387>.
34. Froelich, W. Towards improving the efficiency of the fuzzy cognitive map classifier. *Neurocomputing* **2017**, *232*, 83–93. <https://doi.org/10.1016/j.neucom.2016.11.059>.
35. Nápoles, G.; Leon Espinosa, M.; Grau, I.; Vanhoof, K.; Bello, R. Fuzzy cognitive maps based models for pattern classification: Advances and challenges. In *Soft Computing Based Optimization and Decision Models*; Pelta, D., Cruz Corona, C., Eds.; Studies in Fuzziness and Soft Computing, Vol. 360; Springer: Cham, Switzerland, 2018; pp. 83–98. https://doi.org/10.1007/978-3-319-64286-4_5.
36. Sharma, A.; Tselykh, A.; Podoplelova, E.; Tselykh, A. Knowledge-oriented methodologies for causal inference relations using fuzzy cognitive maps: A systematic review. *Comput. Ind. Eng.* **2022**, *171*, 108500. <https://doi.org/10.1016/j.cie.2022.108500>.
37. Hu, Y.; Guo, Y.; Fu, R. A novel wind speed forecasting combined model using variational mode decomposition, sparse auto-encoder and optimized fuzzy cognitive mapping network. *Energy* **2023**, *278*, 127926. <https://doi.org/10.1016/j.energy.2023.127926>.

38. Mohammadi, H.A.; Ghofrani, S.; Nikseresht, A. Using empirical wavelet transform and high-order fuzzy cognitive maps for time series forecasting. *Appl. Soft Comput.* **2023**, *135*, 109990. <https://doi.org/10.1016/j.asoc.2023.109990>.
39. Salmeron, J.L.; Froelich, W. Dynamic optimization of fuzzy cognitive maps for time series forecasting. *Knowl.-Based Syst.* **2016**, *105*, 29–37. <https://doi.org/10.1016/j.knosys.2016.04.023>.
40. Vanhoenshoven, F.; Nápoles, G.; Froelich, W.; Salmeron, J.L.; Vanhoof, K. Pseudoinverse learning of Fuzzy Cognitive Maps for multivariate time series forecasting. *Appl. Soft Comput.* **2020**, *95*, 106461. <https://doi.org/10.1016/j.asoc.2020.106461>.
41. Chen, C.T.; Chiu, Y.T. A study of dynamic fuzzy cognitive map model with group consensus based on linguistic variables. *Technol. Forecast. Soc. Change* **2021**, *171*, 120948. <https://doi.org/10.1016/j.techfore.2021.120948>.
42. Gan, X.; Yan, K.; Wen, T. Using fuzzy cognitive maps to develop policy strategies for the development of green rural housing: A case study in China. *Technol. Forecast. Soc. Change* **2023**, *192*, 122590. <https://doi.org/10.1016/j.techfore.2023.122590>.
43. Monshizadeh, F.; Reza Sadeghi Moghadam, M.; Mansouri, T.; Kumar, M. Developing an industry 4.0 readiness model using fuzzy cognitive maps approach. *Int. J. Prod. Econ.* **2023**, *255*, 108658. <https://doi.org/10.1016/j.iipe.2022.108658>.
44. Liu, X.; Liu, A.; Chen, J.L.; Li, G. Impact of decomposition on time series bagging forecasting performance. *Tour. Manag.* **2023**, *97*, 104725. <https://doi.org/10.1016/j.tourman.2023.104725>.
45. Hu, Y.C. Incorporating a non-additive decision making method into multi-layer neural networks and its application to financial distress analysis. *Knowl.-Based Syst.* **2008**, *21*, 383–390. <https://doi.org/10.1016/j.knosys.2008.02.002>.
46. Peng, Z. *Fuzzy Cognitive Maps and Their Construction Methods*; Beijing Institute of Technology Press: Beijing, China, 2017.
47. Papakostas, G.A.; Koulouriotis, D.E.; Polydoros, A.S.; Tourassis, V.D. Towards Hebbian learning of Fuzzy Cognitive Maps in pattern classification problems. *Expert Syst. Appl.* **2012**, *39*, 10620–10629. <https://doi.org/10.1016/j.eswa.2012.02.148>.
48. Niu, M.; Sun, S.; Wu, J.; Yu, L.; Wang, J. An innovative integrated model using the singular spectrum analysis and nonlinear multi-layer perceptron network optimized by hybrid intelligent algorithm for short-term load forecasting. *Appl. Math. Model.* **2016**, *40*, 4079–4093. <https://doi.org/10.1016/j.apm.2015.11.030>.
49. Hu, Y.C. Finding useful fuzzy concepts for pattern classification using genetic algorithm. *Inf. Sci.* **2005**, *175*, 1–19. <https://doi.org/10.1016/j.ins.2004.10.002>.
50. Hu, Y.C. Tourism forecast combination using weighting schemes with flow information among component models. *Appl. Soft Comput.* **2025**, *168*, 112498. <https://doi.org/10.1016/j.asoc.2024.112498>.
51. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
52. Xie, G.; Qian, Y.; Wang, S. A decomposition-ensemble approach for tourism forecasting. *Ann. Tour. Res.* **2020**, *81*, 102891. <https://doi.org/10.1016/j.annals.2020.102891>.
53. Hsu, C.I.; Huang, Y.C.; Wong, K.I. A Grey hybrid model with industry share for the forecasting of cargo volumes and dynamic industrial changes. *Transp. Lett.* **2018**, *12*, 25–36. <https://doi.org/10.1080/19427867.2018.1502505>.
54. Hu, Y.C. Forecasting the demand for tourism using combinations of forecasts by neural network-based interval grey prediction models. *Asia Pac. J. Tour. Res.* **2021**, *26*, 1350–1363. <https://doi.org/10.1080/10941665.2021.1983623>.
55. Jiang, P.; Wang, W.B.; Hu, Y.C.; Jiang, H.; Wu, G. Interval grey prediction models with forecast combination for energy demand forecasting. *Mathematics* **2020**, *8*, 960. <https://doi.org/10.3390/math8060960>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.