Research Article

FCTA: A Forecasting Combined Methodology with a Threshold Accepting Approach

Juan Frausto-Solis, Lemuel Rodriguez-Moya, Javier González-Barbosa, Guadalupe Castilla-Valdez, and Mirna Ponce-Flores

Tecnológico Nacional de México/IT Ciudad Madero, Ciudad Madero, Mexico

Correspondence should be addressed to Juan Frausto-Solis; juan.frausto@itcm.edu.mx

Received 29 November 2021; Revised 14 February 2022; Accepted 22 February 2022; Published 25 March 2022

Academic Editor: Samuel N. Jator

Copyright © 2022 Juan Frausto-Solis et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The combination of forecasting methods is a widespread technique. The most common technique for ensembling several individual methods is scoring. These ensemble methods have been useful for designing hybrid forecasting time series methods in several areas. However, more precise applications are required in modern times, and hybridizations using several ranking approaches have emerged to solve this problem. The main difficulty of this technique is finding the most suitable methodology to combine forecasting methods. This work presents a new methodology named FCTA (forecasting combined method with threshold accepting) for ensembling several forecasting methods. This methodology uses a Threshold Accepting algorithm for weighting individual predictions. FCTA starts from an initial weighting and aims to find the best ponderation of the individual methods by optimizing the precision of the global prediction. For testing FCTA, we selected a dataset taken from M4-Makridakis-competition, and we compared it with the best individual forecasting methods. FCTA is also compared with other successful methodologies. The experimentation shows that FCTA surpasses the best M4 individual methods and is equivalent or better than the best methodologies of the area.

1. Introduction

Forecasting is a complex problem, and for a long time, the forecasting community has focused its research on finding better techniques by minimizing the prediction error. Nowadays, the common methodologies apply filters and ensemble the best individual methods in many areas [1–7], and among them, the two with the best performance are PP-MA [1] and LA [2] methodologies. The former uses a filtering stage and the Bayesian information criterion (BIC) to select the best forecasting models for each slot of the time series [1]; the second, uses a novel learning automata approach deriving two main ensemble algorithms named LA-SW and LA-MW [2]. Other ensemble methods are EMD-PE-ANN and LSTM-KF applied to wind speed and energy consumption, respectively [4, 5]. Combining multiple forecasting methods involves selecting them and determining their best weights distribution [8–13]. An ancient method is scoring; it combines several forecasting methods [14–15]. On the other hand, selecting a single prediction method can be seen as a risk when the performance of the models and data vary over time [9], and it is better to use two or more individual methods in a combination. The Bates–Granger ensemble methodology (B&G) has strong foundations [16] and has inspired successful modern approaches [13, 17], including the methodology proposed in this paper.

One of the enormous challenges in forecasting is to develop a method with high requirements: a large spectrum, well-performance in multiple scenarios, datasets from several areas (industrial, financial, and so forth), and for diverse forecasting horizons such as one period or \( n \) periods ahead. The motivation to develop the proposed methodology is to solve this problem. The M4 competition permits these scenarios for testing new methodologies, as is proposed in this paper. As a result, the M4 has become a benchmark and
includes the forecasting results for one hundred thousand series, with the best prediction methods [18].

In this paper, we present the FCTA methodology (forecasting combined method with threshold accepting), which combines several forecasting M4 methods. The significant FCTA elements are its architecture and the TAE algorithm (threshold accepting-enhanced). TAE is an innovative optimization method that uses a reheat and golden ratio heuristics to improve exploration of the search space. The experiments using almost twenty thousand times series and M4 methods show that FCTA obtains better forecasting results than all the individual methods in the combination. Besides, FCTA has statistically equivalent or better performance than two of the best methodologies of the area. The paper is organized as follows. Section 2 contains a general description of FCTA and the metric error used to evaluate the performance of the forecasting techniques. In Section 3, we present related works about combining forecasting methods. Section 4 presents a detailed FCTA description, architecture, and TAE algorithm. Section 5 presents the experimentation process, dataset description, results, and analysis. Finally, in Section 6, we present our conclusions.

2. Materials and Methods

Despite many publications in the forecasting area, new methods are still necessary to predict future time series values more accurately [19]. The forecasting methods have mainly used two approaches. (1) Using the best individual algorithms, and (2) Combining several approaches (hybridization). The latter has become the most popular in the last years. This work is related to both approaches. Makridakis-Competitions provides times series, which are currently used as a benchmark for evaluating the quality of forecasting algorithms. M4-Competition, in particular, provides one hundred thousand time series for this task. This section presents a general description of FCTA Methodology, which is tested using datasets from M4-competition.

2.1. Combination of Forecast with FCTA. Combining several forecasting methods aims to obtain better accuracy than each method in the combination; individual execution models provide forecast values which are combined to obtain the final forecast [20]. A key issue is the adequate number of combined methods to produce the lowest prediction error. The best practice is to combine methods based on their individual performance [20, 21]. The FCTA process is shown in Figure 1 where $x^k_t$ is the time series $k$ which the last element is in the period $t$. For instance, the dataset \{x^1_{1}, x^2_{1}, ..., x^6_{2}\} has the six time series: F. In other words the set \{x^1_{1}, x^2_{2}, ..., x^6_{7}\} represents the data in the period 1, 2, ..., $t$ for the $k$th element of this dataset.

Our methodology requires selecting the best $k$ methods for any dataset of the area. The selected individual methods are named in Figure 1 as M1, M2, ... Mk, and $f_1, f_2, ..., f_k$ representing the predictions of these forecasting methods. Then FCTA obtains $k$ estimations in $h$ periods in the future, which are represented by $\hat{Y}_{t+h}(M1)$, $\hat{Y}_{t+h}(M2)$, ...

$\hat{Y}_{t+h}(Mk)$. The weight estimations process is described in detail in section 4.2.3. This process obtains a $\omega_j$ weight for each method. These weights are averaged in a combination process of Figure 1 to produce the initial forecasting for each time series. Nevertheless, this prediction does not have, in general, a better prediction than the individual methods. Thus, a refined algorithm, named TAE (Threshold Accepted Enhanced) based on the threshold accepting algorithm [22], is applied and presented in another section. The combined forecast $\hat{F}_{t+h}$ can be obtained as a function of the $k$ individual predictions of the best methods, where an initial ponderation $\omega_0^k$ is assigned. Nevertheless, obtaining this function is not, in general, an easy task, although a simple weighting function for pondering the individual predictions is commonly used. On the contrary, as mentioned before, in FCTA, we refine the $\hat{F}_{t+h}$ prediction, which is done by the TAE algorithm.

2.2. Forecasting Metrics. The error metrics are used to estimate the quality of the forecasting methods; the most common are MAPE, MASE, and sMAPE [23–25]. The sMAPE metric is a variation of MAPE commonly used in M4-Makridakis competition; thus, we use this metric which formula is shown in equation (1)

$$s\text{MAPE} = \frac{2 \times \sum_{t=1}^{n+h} |Y_t - \hat{Y}_t|}{n \times (|\bar{Y}_t| + |\bar{\bar{Y}}_t|)} \times 100\%,$$

where $Y_t$ is the current observation of the time series; the variable $\hat{Y}_t$ represents the predicted value, $n$ is the amount of data in the time series, $m$ is the time interval between successive observations considered, and $h$ represents the length of the forecast horizon.

3. Related Work

Combining forecasting using weights was proposed by Bates–Granger [16], as in our model, weights are established for all the individual methods. The weights are determined according to the variance of the errors of the individual methods [14]. B&G determines the weights using the correlation between the variance of the errors of the individual forecasting methods in the combination. This method can be applied to several methods in the combination. When there are only two methods, weights $\omega_k$, and $1 - \omega_k$ should be assigned to the first and second method, respectively. The combined method with this strategy does not necessarily produce a better prediction than the best individual method. However, these weights should be modified to enhance the combined forecasting. B&G established that the best combination of individual methods is obtained when [16]:

(a) The combination produces the lowest error

(b) The total variance of the error is as low as possible

The above methodology was applied many times for the one-day-ahead [2, 5] and larger forecasting horizon [4, 14, 26–30]. Some methods are relatively easy to be implemented, but most of them require adjusting their
parameters for different applications [1, 2]. Furthermore, a linear combination does not necessarily surpass the prediction of the best method in the combination [31]. As a result, the forecasting community proposed many alternatives [18]: (a) determine the weights using Monte Carlo Methods with a distribution density probably defined by the variance matrix of the combination, (b) to use only the diagonal variance values as a factor for obtaining enhanced weights. Some authors have used the latter ideas for large datasets. Specifically, in [9], the authors used weighting ponderation in time series taken from the competitions NN3 [3] and NN5 [32]. According to the experiments with enhanced weight, a pool of methods obtained better results than individual methods [9]. Also, the authors apply clustering and classification, and they use a large set of features in a decision tree. They determine which features are essential for selecting methods in the combination. We note that the best combination is usually obtained when the best individual methods are used and when the combination has the most diverse approaches [2, 3]. For example, Jaganathan and Prakash propose two forecasting combination approaches methods based on [33]: (a) historical evidence of well-evaluated methods inspired in Uniform weight distribution [34] and (b) weights optimization [33]. Jaganathan and Prakash took 24 forecasting methods from several sources to design a combination that surpassed most of the M4-Competition methods. Table 1 shows some of the most relevant related works regarding the used technique, performance metrics, benchmark, advantages, and disadvantages of the work.

4. Threshold Accepting

This section reviews the TA technique and its extension TAE within the FCTA methodology.

4.1. Basic Threshold Accepting. The threshold accepting (TA) algorithm proposed by Dueck–Scheuer has been applied to solve NP-hard optimization problems from different areas. TA is a variation of simulated annealing and, likewise, represents an analogy of a thermodynamic process, and it has the following elements. (Algorithm 1):

(a) The objective function \( f(x) \): the metric error in our forecasting problem.
(b) The neighborhood structure: it defines how to move from a current solution to a new one. The movements are defined by the function \( \text{Generate} \) (line 6).
(c) The deterioration criterion: it defines when accepting wrong solutions. Simulated annealing used the Boltzmann criterion. Instead, TA simplifies that, using a range of deterioration of the objective function (line 7).
(d) The stop criterion: uses the difference between two neighbor solutions, \( \Delta \) in lines 6 and 7), and the current Temperature. It is represented as the rule in line 11.

In the TA algorithm, the objective function proposes to find the closest solution to the optimum. TA randomly seeks a new solution in the search space, accepting it when the objective function is enhanced. This algorithm consists of two loops:

(i) The external loop: it is controlled by the temperature parameter that varies from the initial to the final temperature. The former is too high (for instance 100, or 1000), and the latter is too small (close to zero).
(ii) The internal loop: this loop commonly increases the number of iterations with a fixed or variable rate, called beta in this paper.

4.2. A Detailed Description of FCTA Methodology. FCTA is a methodology for calculating the best weight distribution of individual forecasting methods into a combination of forecasts that minimize a metric error.
4.2.1. Forecasting Optimization Model. The problem of obtaining the best combination of $M$ individual forecasting methods for the period $t + h$ can be mathematically described by the optimization model of

$$\min_{\omega} \bar{E} = \sum_{i=1}^{M} \omega_i \bar{Y}_i,$$

Subject to $\sum_{m=1}^{M} \omega_j \geq 1; \forall \omega_j \geq 0; I = 1, \ldots, M$.

where $\bar{E}$ is the estimated error for the combined forecasting methods; $\omega_j = (\omega_1, \ldots, \omega_M)$, and.

\(\bar{Y}_i = f(M, \bar{Y}_{t+h})\) gives the forecast for the period $t + h$ of the $M_i$ method. All the individual methods predict $h$ periods in the horizon whose size depends on the dataset (Yearly, Quarterly, Monthly, Weekly, and Daily). Time series used in this work were taken from the M4-Makridakis competition [17].

4.2.2. FCTA Architecture. The architecture FCTA methodology is described in Figure 2, which was designed for testing FCTA for any application area of the M4 competition. The forecasting methods are previously executed for all

| Related work | Technique | Metrics | Benchmark | Advantages | Disadvantages |
|--------------|-----------|---------|-----------|------------|---------------|
| [1] PP-MA    | MAE, RMSE, MAPE, MASE | PlanetLab | Easy to implement | Lacks optimization to enhance the combination |
| [2] LA       | Rmsd, Error ratio, absolute error | PlanetLab | Easy to implement | Lacks optimization to enhance the combination |
| [4] EMD-PE-ANN | MSE, RMSE MAE | Wind speed data | High performance in larger horizons. It can be used in real-time | Once training is readjusted for future data. Is required |
| [5] LSTM-kalman filter | MAPE, MSE, RMSE, and R2 score | Residential electric consumption | Forecast for dynamically changing systems | Forecast only one period ahead. Tuning for new applications is required. |
| [13] FFORMA | sMAPE, OWA, MASE | M4 Competition | High performance | Does not permit to select specific individual methods |
| [14] CUSUM | SRMSE | Monthly demand data for ten items | Detects if the weights should be adjusted for new applications | Precision depends on its ARL parameter that should be adjusted |
| [16] B&G | MSE | Airline passenger data | Good performance in small horizons | Poor performance for large horizons |
| [33] Jaganathan | sMAPE, OWA, MASE | M4 Competition | Better point forecasts, better confidence intervals. | Does not permit to select specific individual methods |
| Proposed FCTA | sMAPE, ErrorRatio | M4 Competition | Well-performance to forecast in multiple scenarios and large horizons | Its tuning process requires specialized knowledge |

MAE, mean absolute error; MAPE, mean absolute percentage error; MASE, mean absolute scaled error; RMSE, root-mean-square error; SRMSE, square root mean square error; MSE, mean-square error; EMD-PE-ANN, empirical mode decomposition-permutation entropy-artificial neural network; LSTM, long short-term memory; sMAPE, symmetric, mean absolute error; OWA, overall weighted average.

\textbf{Algorithm 1: Classical threshold accepting.}

(1) \textbf{Initialize}: $n_{\text{steps}}$, $\alpha$, convergence = false; $k = 1$

(2) Compute threshold sequence $T_k$. Compute $f(x^{\text{old}}) = f(x^*)$.

(3) Randomly generate a current solution $x^{\text{new}} \epsilon \mathbb{X}$

(4) \textbf{while} convergence = false \textbf{do}

(5) \textbf{for} $i = 1: n_{\text{steps}}$ \textbf{do}

(6) Generate $x^n \epsilon \mathcal{N}(x^*)$ and compute $\Delta = f(x^{\text{new}}) - f(x^{\text{old}})$

(7) \textbf{if} $\Delta < T_k$ \textbf{then} $x^{\text{old}} = x^{\text{new}}$

(8) \textbf{end} for

(9) $k = k + 1$

(10) $T_k = \alpha T_k$

(11) \textbf{if} $\Delta \leq e$, or $k \geq N_{\text{max}}$ \textbf{convergence} = true.

(12) \textbf{end} while

(13) $x^{\text{old}} = x^{\text{old}}$
the time series, and their errors are stored in a dataset named SFM (Set of Forecasting Methods).

First, we have two databases: (a) the forecasting methods of the area and (b) a dataset with the time series requiring a forecast. As is typical, each time series is divided into three sections: Training, Validation, and Testing. For each time series, the process Selection determines the quality of the methods ranking them from best to worst. This ranking is stored by Selection in the Dataset as attributes of each time series. Thus, the best methods for each time series are determined; also, the average error and standard deviation for each time series are calculated in this process.

Weighting is a process that calculates an initial weight for each k method. This process uses two strategies: Uniform (FCTAU) and Standard Normal (FCTAN) distributions. These weights are calculated and stored in the Xo vector, which is an initial solution for the TAE algorithm. The TAE process determines the best ponderation for each method. Both Weighting and TAE are explained in other section.

Finally, TAE obtains the optimal weight distribution to determine the Combined Forecast of the selected best methods. This final forecast is computed using the Testing part of the time series. Then the error metric for the final forecast is obtained, and the process is finished.

Figure 3 shows the sequential diagram for FCTA, which order the actions as is described as follows:

(i) The actor asks “Selection” to start the process with some options. Then, “Selection” requests to DataSet for the subset of time series (subdivided into training, validation, and test data), which are returned to it; “Selection” also demands SFM for the metric errors, which are returned to it.

(ii) Selection” sorts the best k methods and sends them in a priority vector with their initial weights. “Weighting” uses one of two distributions (Normal or Uniform) to obtain an initial weighting vector named Xo and sends it to TAE; Xo has the first ponderation of the methods.

(iii) TAE obtains the data for validation from the DataSet, with Xo and the TAE optimization process enhances the ponderation. Then, TAE obtains the forecasting values and metric errors for the testing part. Finally, TAE sends the results to the actor.

4.2.3. Selection of the Best Methods. FCTA selects the best methods for each time series with the process “Selection” of Figure 2. This process works as follows:

Step 1: the performance of each method (according to the metric error) in the validation set validation is previously stored in the SFM repository.

Step 2: the individual methods of Step 1 are ranked according to their metric errors for each time series. The order is stored in SFM in a ranking vector. The methods in this vector are named BestM1, BestM2, . . . BestMK, . . . BestMN.

Step 3: the best N methods, with the lowest metric error, are chosen by Selection process.

To include several methods in the combined forecasting, Selection uses the Include Process shown in Algorithm 2. First, this algorithm starts (line 1) with a Ranking vector which has the best N methods arranged in priority order (BestM1, . . . , BestMK, . . . BestMN). In line 2, the algorithm selects the two best methods (BestM (1), BestM (2); this list is increased if the subsequent best method is not worse than the latter in the list in a 𝜙 small value previous tuned (line 6); if this condition is accomplished a new best method is added to this list, and the number of the best method is increased by one (line 7); the latter method in this list is consistently identified as the best current method in line 8 or BestM (current); when the last condition (in line 6) fails the Boolean variable named include takes the false value in line 9 and the process is finished. Include Process continues selecting a method until the stop criterion (include = false) is applied. This criterion depends on the threshold 𝜙 defined in the tuning process.

To find the best weights for the combined forecast, we define a vector of weight Xo = (ω1, ω2, . . . , ωj, . . . , ωm). This vector is an initial solution for pondering the forecasting methods ((1, 2,&, m)) in the combined forecasting. The process Weighting in Figure 4 uses the Uniform and Standard Normal distributions for defining this Xo vector. For the first case, the process simply assigns the same weight to each method. The second case is explained in the next paragraph.

Figure 4 shows that the Weighting process receives for each time series the following information previously generated: the forecasts and metric errors. Furthermore, this process reads the average and standard deviation of these metrics associated with each group of time series. For the Standard Normal distribution: a population of metric errors is used for each group of time series, where several samples of metric errors are taken. All the samples are characterized by their average error. As is well known, when the number of samples is large enough, they follow the Standard Normal distribution. Then Weighting determines the mean μ of these average errors and its standard deviation σ. At this point, it uses the Standard Normal distribution to normalize the metric errors by using Z-score values determined by the well-known relation Zc = ci − μ/σ; where, ci is the average of the forecast error of the method Mi; μ is defined as the mean of all the average errors methods in the entire dataset; and σ is the standard deviation of all the Zc values area under the normal distribution curve is easily determined.

By definition, the total area under the standard curve represents around 100% of the probability related to this distribution, as shown in Figure 5. Therefore, in our case, we need to find out what percentage of the probability represents the area under the curve for each average error to obtain the partial weight. Thus, we apply formula (3) for obtaining that we name the partial weights for each method Pωj. They are named partial weighths, because they are not
1: Request start process
3: Request methods errors
4: Return methods errors
2: Request Train and Validation ts
5: Request Train and Validation ts
6: Order and Include
7: Send best k forecasting methods
8: Initial weighing Solution
9: Request Validation ts
10: Return Validation ts
11: Request Test ts
12: Return Test ts
13: Results

Figure 2: FCTA methodology.

Figure 3: Sequence diagram of FCTA methodology.
the final weights; this is because, in general, they do not satisfy the constraint in the optimization model defined by equation (3).

$$Pw_i = 1 - \frac{A_i}{\sum A_i} \quad (3)$$

where $A_i$ represents the area under the curve for each method in the combined forecasting, and $\sum A_i$ represents the sum of all the areas under the curve for these methods. In formula (3), the number one on the right-hand side represents the total area under the standard curve. In Figure 5, we give an example, where the area under the curve is divided in $1\sigma$, $2\sigma$, and $3\sigma$; it is obvious that $\sum Pw_i$ is different from one. The final process consists of determining the final weights $Fw_i$. First, having the partial weights, we normalize them by $Fw_i = Pw_i/\sum Pw_i$. On the other hand, the initial weights can be determined by replacing the normal distribution for the Uniform distribution, and a similar process is followed. Then, Weighting process provides an initial solution to the TA algorithm.

**Algorithm 2: Include Process.**

4.24. TAE Algorithm. The proposed TAE algorithm is shown in Algorithm 3, and some of the strategies incorporated in this algorithm: roulette, memory, cooling scheme, reheat, and the thermal equilibrium to be explained in subsequent sections.

TAE uses the parameters: $T_{initial}$, $T_{final}$, $\alpha$, $L_k$, $\beta$, $X_0 = (\omega_1, \omega_2, \ldots, \omega_j, \ldots, \omega_m)$, $E_0$, $F_0$. The variable $X_k$, contains the weights of the method in the combination and is provided by the Weighting process (line 1). First, the following parameters are initialized $T_k$, $X_{old}$, $\alpha$, $\Phi$ (lines 2-3). The temperature parameter controls the external cycle (lines 4-33), while the internal cycle (lines 6-28) by the number of iterations (which in our case is variable); the parameter $L_k$ determines this number of iterations. The first procedure is to generate a new feasible solution using the initial solution $X_0$ stored in $X_{old}$. A random number is used by the Roulette and Memory methods (lines 8 and 10). The new solution $X_{new}$ is determined by applying a perturbation function to $X_{old}$. Then, an energy calculation is performed (line 12). TAE obtains the energy $E_k$ with the sMAPE, $E_0$ with $X_0$ and the subsequent with $X_k$.

The ensuing process in the sequence is the $\Delta E$ calculation (line 13), which is the difference between the energy of the current and candidate solution. Therefore, we subtracted the error of the current combined forecast from the new one (candidate solution). At this moment, if $\Delta E < 0$ (line 15) (the combination MAPE decreases), this new solution is accepted $X_{old} = X_{new}$ (line 16). Moreover, the process of generating a new solution started again; otherwise, if $\Delta E > 0$ and if $\Delta E \leq Tol_k$ (line 21), this new solution is also accepted $X_{old} = X_{new}$ (line 22). Then, the process of generating a new solution starts again. If $\Delta E > Tol_k$ the process continues, if $k < L_k$, $k$ is increased (line 27), generating a new solution, starting again, and iterating a local search. On the other
hand, if $k \geq L_k$, the inner cycle ends, and the parameters are updated: the cooling speed, ($\alpha$ parameter) in line 29 and a first final temperature with the $\Phi$ parameter; in line 30, the temperature with $T_{k+1} = \alpha \cdot T_k$, the threshold tolerance $\text{Tol}_k$, and the $L_k$ parameter with the $\beta$ value (line 30). The reheating strategy is applied when stagnation is detected. In the external cycle, the slope is calculated to activate the stop criterion is activated in line 32. Lastly, the process is finished when the stop criterion is accomplished. The TAE main processes are described in Section 4.2.5.

The variables and parameters in the TAE algorithm are as follows:

(i) The initial parameters: $T_{\text{initial}}$: initial temperature; $T_{\text{final}}$: final temperature; $\alpha$: cooling rate; $L_k$: number of iterations of the inner cycle; $\beta$: search increase rate; $X_o$: initial solution (weights); $E_o$: initial energy (sMAPE combination).

(ii) Control parameters: $T_k$: current temperature; $T_{\text{min}}$: minimal temperature to stop algorithm; $\text{Tol}_k$: current temperature tolerance; $\gamma$: tolerance rate; $k$: current iteration; $\Phi$: golden ratio number; $\Delta E$: difference between current and previous energy; $\text{NonImprove}$: number of iterations without improvement; $\text{MaxX}$: maximum iterations without improvement; elitism value; $\text{MaxE}$: maximum elitism value; $\text{Minc}$: minimum slope curve value.

(iii) Solution variables: $X_{\text{old}}$: current solution; $X_{\text{new}}$: new solution; $E_{\text{old}}$: current energy; $E_{\text{new}}$: new energy; $T_f$: temperature window; $\text{slope}$: curve slope; $F_{10}$: forecasting values of the selected methods.

4.2.5. Generating a New Solution in TAE. The new solution is generated following one of the two methods: the roulette and memory methods. They are explained in the following paragraphs.

Roulette Method. Having an $X_k$ vector or weight configuration, the subsequent vector $X_{k+1}$ is determined by the Monte Carlo method as follows:

Firstly, the process starts with the initial solution $X_0$ which is a vector of the weights in the combination $X_0 = (\omega_1, \omega_2, \ldots, \omega_i, \ldots, \omega_m)$. TAE determines the intervals for the weights in the initial solution $X_0$, and subsequent solutions. For instance, for five methods $X_0 = (0.30, 0.15, 0.20, 0.25, 0.10)$, the five intervals are defined with the ranges: $0 < \omega_i \leq 30; 30 < \omega_i \leq 0.45; 0.45 < \omega_i \leq 0.65; 0.65 < \omega_i \leq 0.90; 0.90 < \omega_i < 1.0$.

Secondly, the next process consists of generating a new $X_{k+1}$ solution; due to the condition that the summation of the weights should be one, the new weight must satisfy this condition. Thus, if the algorithm removes a portion (quantum) of weight from one method, it should compensate this portion on another weight. Two random numbers are generated in the range [0, 1]; the first identifies the method for adding a value, while the second corresponds to the method for subtracting this value (Figure 6).

Memory method. In this case, the algorithm has a structure named "memory" that stores the algorithm’s last five iterations movements.

This structure allows verifying in the last movements those methods that have enhanced the metric error for the optimal solution. As well as in the roulette method, the process starts with a solution. $X_o = (\omega_1, \omega_2, \ldots, \omega_i, \ldots, \omega_m)$.

Firstly, an average of the last five movements is calculated and stored in the memory. Secondly, with the average previously computed, it is evaluated which method is beneficial for adding a quantum of weight and which method is convenient to decrease a quantum value of weight.

4.2.6. Energy Calculation in TAE. The TAE algorithm is analogous to the classical Golden Ratio Simulated Annealing (GRSA), which minimizes an energy variable at each temperature [35]. Usually, the algorithm determines this energy with the current state solution, and at the beginning of the process, it commonly has a high value. At each step, with a new solution, its new energy is determined. The energy value of a solution determines its probability of being accepted as a current solution. When the temperature is high, the algorithm accepts almost all the new solutions, even if they have a high error. When the temperature is low, the algorithm accepts only a few solutions in which energy has not been decremented. As in TA algorithms, the energy defines the search space of the next iteration. As the energy variable is reduced, the search space is reduced too. A new energy value is evaluated in TAE with a metric error commonly used in M4 forecasting evaluations: the sMAPE. In this case, the combined forecast value is given by

$$F_{10} = X_{\text{new}} * (f_{11}, f_{12}, \ldots , f_{1n}), \quad (4)$$

where $X_{\text{new}} = (\omega_1, \omega_2, \ldots, \omega_i, \ldots, \omega_m)$ has the new weights for pondering the individual forecast $f_{11}, f_{12}, \ldots , f_{1n}$ for the $n$ chosen methods.

In TAE, the difference between the current and new energy permits the algorithm evaluation if the current iteration produced a good or bad movement in the optimization process.

4.2.7. Cooling Scheme. TAE can be improved using the Golden ratio (GR or $\Phi$) technique. In a search process, GR is a parameter used to divide the searching space into several GR sections; in Simulated annealing and Threshold heuristics, GR is a parameter for dividing the temperature cycle; each section has different initial and final temperatures [35]. The search strategy used by GRSA performs faster searches in every GR section. This $\Phi$ constant (or aura parameter) divides the temperature interval into segments, and in this way, the search space is faster explored. We used the geometrical cooling scheme, $T_{k+1} = \alpha \cdot T_k$, where $T_k$ and $T_{k+1}$ are the current and new temperatures, respectively. The $\alpha$ parameter is known as the cooling factor, and it is in the range $0.7 \leq \alpha < 1$. In the first GR, $\alpha$ is small ($\alpha = 0.7$), then the temperature decreases rapidly. In contrast, in subsequent
GRs, $\alpha$ is growing, and the explorations are more and more slowly; the last GR section has an $\alpha$ value close to one, and the acceptation criterion becomes very strict. This cooling factor variation allows decreasing the temperature more slowly when the temperature is closer to the final temperature. The temperature $T_\phi$ is modified, using the golden

Algorithm 3: Proposed Threshold Accepting Enhanced (TAE) for FCTA.

| Methods | M1 | M2 | M3 | M4 | M5 |
|---------|----|----|----|----|----|
| Weight  | 0.30 | 0.15 | 0.20 | 0.25 | 0.10 |
| Intervals | 0–0.30 | 0.30–0.45 | 0.45–0.65 | 0.65–0.90 | 0.90–1 |

Pac: Cumulative density probability
Ran1: First random number for increasing a $\varphi$ quantum
Ran2: Second random number for decreasing a $\varphi$ quantum

Figure 6: Example of MC for decreasing and compensating quantum weights.
number \( \Phi \), the search space to find the optimal solution is reduced when the GR strategy is used to define the cooling scheme. Figure 7 shows the sections where the rates of temperature decrease change, thus decreasing the number of iterations of the algorithm to arrive at the optimal solution.

4.2.8. Reheat Strategy for TAE. Another strategy used in TAE is Reheat (RH) [36]; in RH, the current temperature \( T_k \) is restarted at an earlier value if the algorithm does not improve its quality during a determined number of metropolis cycles. The algorithms restarts the temperature to explore more solutions from the search space. In TAE, the RH is applied in the last GR section, and when detects stagnation condition; this is performed when for some iterations (ten is common), a significative improved was not achieved (MaxY = 10 in lines 3 and 5 of Algorithm 3). The RH process is started when this number is reached (line 5 of Algorithm 3). Then, the RH process is repeated a maximum number of times (commonly three times). After this number, TAE is normally executed until the stop condition is reached. The HR strategy permits the algorithm to escape from local optima, explore more solutions, and avoid premature convergence. Figure 8 shows some elements of the RH strategy.

4.2.9. Thermal Equilibrium in TAE. The dynamic equilibrium of a TA algorithm is the metric that expresses when the current temperature \( T_k \) has reached a value very close to zero, indicating that the search is finished. The dynamic equilibrium in the TAE algorithm is achieved when two conditions are satisfied: \( T_k < T_{\min} \) or when \( \text{slope} < \text{Min} \) \( \text{slope} \). Therefore, the

In Figure 8, reheat algorithm calculates the slope of the curve in each iteration. The minimum slope curve value is \( \text{Min} \) which indicates that the thermal equilibrium has been reached. At this point, the end of the search is determined.

4.2.10. TAE Complexity. As is shown in Figure 3, the FCTA execution time depends essentially on its elements selection, weighting, and TAE, which complexity classes for the worst-case are the following:

For selection: \( O(k \log k) \) is due for sorting the \( k \) best methods, and \( O(k) \) for selecting the best methods that will participate in the combination.

For weighting: \( O(k) \) for pondering the \( k \) methods, and \( O(1) \) for storing the weights in the priority vector and sends it to TAE. Thus, complexity of Weighting is \( O(k) \).

For TAE: \( O(n^2 + n) \log n \), where \( n \) is the number of temperatures, that is the complexity class for the threshold accepting algorithm, even for the multi-objective Case [37, 38].

FCTA is based on pondering methods as B\&G, which performs similar tasks, except that the latter does not have an enhancing stage as TAE. In other words, TAE provokes an overhead which complexity is after the big-O reduction: \( O(n^2 \log n) \).

5. Experimentation and Results

This section evaluates the combined forecast performance using the weight distribution with the FCTA methodology.

5.1. Dataset Description. The dataset used corresponds to some of the M4 competition series. The Makridakis M4 Competition contains one hundred thousand time series. It is built from different public domain sources, such as industries, services, and exports, [17]. The time series are divided into six groups with different frequencies: Yearly (Y), Quarterly (Q), Monthly (M), Weekly (W), Daily (D), and Hourly (H). Due to the extensive length of this dataset, the time series can also be divided into six types: demographics, finance, industry, macro, micro, and other. The criteria for selecting a representative subset of this data for testing require at least two hundred fifty until one thousand observations and are shown in Table 2.

In each frequency, the time series have different types. Therefore, the hourly group was not selected because it contains only time series of type Other. At the end of this table, there are two final columns named size and percentage (%). The first refers to the number of selected series in each frequency group. The last column is the corresponding percentage of time series selected relative to the total.

5.2. Experimentation Environment. The Experimentation Environment used to run the FCTA methodology is done using the R language. The hardware configuration used for processing all FCTA components is displayed in Table 3.

5.3. Experiment I: FCTA versus M4 Competition Methods. The experiment consisted of applying the proposed FCTA methodology and forecasting the future values of the selected time series of the M4 Competition. We define two scenarios: Scenario A. Individual forecasting methods in the combination compared to FCTA.

Scenario B. Performance of FCTA for different frequencies of the dataset (Y, Q, M, W, D).

The experimentations is explained as follows:

(i) The experiment combines the best forecasting methods to forecast future time series values using one of the two weight distribution techniques (Uniform or Standard Normal) (Figure 2), which is optimized with the TAE algorithm.

(ii) Each time series of the selected subsets of the M4 is divided into three parts: Training, Validation1, and Validation2, as shown in Figure 9. The test portion described in the figure is unknown to the algorithm, as it is common to corroborate the final forecast.

(iii) First, the Training portion of each time series is used to forecast the Validation1 portion, with all the individual forecast methods available. With the results obtained, the best methods for forecasting the portion of the Validation1 are determined. The
results at this point also define the distribution of weights for the combination of these best methods.

(iv) Next, with the portion of Validation1, Validation2 is predicted, and the best methods are obtained, with the weight distribution as previously calculated. The result is stored, and the optimization process with TA starts using the portion of Validation1 to forecast Validation2.

(v) The weights are adjusted until there is no further possible improvement.
Finally, the best combination of weights and the best forecasting methods are applied using the historical time series to forecast the Test portion.

The fourteen methods used in the experiment are described in Table 4. The first twelve methods are available in [39] while the last two methods, FFORMA [13] and Jaganathan [33] are in M4-Competition.

The optimization process uses the proposed TAE algorithm, which we observed in Algorithm 3. As shown in Table 6, the results of this experimentation present all the 14 individual methods, including [13, 33], used for the ensemble.

The results of the above methods are contrasted with the proposed methods’ results: FCTA and Bates Best (BAB). We designed the latter based on the classical B&G methods. The results indicate that, on average, FCTA has a lower average error (despite the weight distribution chosen) than the best individual method for each series in all of the subsets used. However, in the monthly subset, the difference between the average error of the best individual and FCTA is very little. Because of this, to validate the obtained results, two statistical tests were applied to them: the Friedman and Wilcoxon tests.

Three plots have been developed to view the results from another perspective. The first plot (Figure 10) shows a comparison between FCTA using a Uniform weight distribution (FCTAU), and FCTA using a Standard Normal weight distribution (FCTAN), and the reference methods. Then, in Figure 11, the three proposed methods (BAB, FCTAU, FCTAN) have been compared. Finally, in Figure 12, we have selected the three best methods from all fourteen to compare them with FCTAN, FCTAU, and BAB.

We observe in Figures 11 and 12 that the BAB method increases its metric error when the horizon is higher than fourteen. Thus, we recommend using BAB method only when the horizon prediction is small. On the contrary, FCTAU and FCTAN have consistently obtained a good performance, considering them in the following experiments. The analysis of FCTA in the two scenarios produced the following observations: (1) FCTA produces better results than the forecasting methods in the combination; (2) FCTA produces excellent prediction results for all the frequencies in data sets (Y, Q, W, D) except for monthly series.

5.3.1. Statistical Tests. Friedman’s nonparametric hypothesis test seeks to test the differences between groups (three or more paired groups) when the same parameter has been measured under different conditions on the same subject. Wilcoxon, also a nonparametric test, seeks to find significant differences between two specific groups. Since Friedman’s test does not give the average range for each group, the idea is to run the Friedman test using all the individual and proposed methods, and then run the Wilcoxon test with the top two frequencies. The results of statistical tests using Standard Normal weight distribution are shown in Table 7.

According to Table 7, Friedman’s test indicates that with 99.9% certainty FCTA is the best method compared to all individuals in Y, Q, M, W, and D subsets. To verify how significant the difference is between the best and the second-best forecast methods, we proceed to perform the Wilcoxon test. The results indicate that the FCTA is statistically equivalent to Bagged for the Y set in this second test. However, in the other subsets, FCTA is statistically superior to the second with 99.9% certainty. Also, we can observe the results of the statistical tests when we use Uniform weight distribution. Friedman’s test indicates that with 99.9% certainty FCTA is the best method compared to all individuals in Y, Q, M, W, and D subsets. The Wilcoxon test was executed to verify the best and the second-best forecast.
difference. The results of Wilcoxon test indicates that FCTA is statistically superior to the second, with 95.2% certainty in the Y set and 99.9% of certainty in the Q, M, W, and D subsets.

The results have been plotted in box plots to visualize the difference between the 1st and 2nd methods in all subsets. For example, Figures 13 and 14 (Q and D frequency) show a smaller sMAPE error variability and the smaller range error of the FCTA variants.

5.4. Experiment II: FCTA vs. LA-SW, LA-MW, and PP-MA. The experiment contrasts FCTA with current state-of-the-art methods of combining forecasts; five of the 14 individual methods are used to combine forecasts AA, ET, HO, TH, BA (explained in Table 4). The methodologies comparison are FCTA (with FCTAU, FCTAN); LA (with LA-SW and LA-MW) [3]; PP-MA [2]; and BIC (with BICU and BICN). BICU and BICN are methods that make the selection of the individual methods that will be combined according to the measure of the BIC and optimize the weights assigned to each of these methods using the proposed Threshold Accepting Enhanced algorithm (TAE) In this experimentation, only LA algorithms do not use TAE algorithms for enhancing their weights. The experiment uses the first 30-time series of each group obtained by the filtering process shown in Table 2. This experiment evaluates the forecast combinations strategies in terms of the quality of the final solution and the time to obtain it. Each combination methodology is executed using the same set of individual methods and the same set of time series. The differences consist of how the methodologies explore the time series and combine the individual methods. The experiment consisted of testing for each methodology its one-day-ahead forecast and the n-day-ahead forecast (determined with the same individual methods). The sMAPE error average results for the 150 time series used in the analysis are shown in Table 8. The first row presents the results for the one-day-ahead horizon, while the second row presents the results for the n-day-ahead horizon. Finally, the final row shows the average execution time in seconds to process the 150 time series.

The results presented in Table 8 indicate that, on average, FCTAN has a lower average error than the other ensemble prediction algorithms in both forecast horizons (H1, Hn), followed by FCTAU in H1 horizon. In Hn, the second and third places are for FCTAU, and BICN, respectively.

Concerning execution times, the different FCTA variants have a shorter time than the LA variants and PP-MA. To validate the results, two statistical tests were applied: the Friedman and Wilcoxon tests shown in Table 9.

Table 9 shows the Friedman and Wilcoxon tests, where the first column shows the horizon, which can be 1 or n. The second and third columns show the result of the Friedman test, the methods ordered from best to worst, and the percentage of certainty that there is a statistical difference among those methods. Finally, columns fourth and fifth show the result of the Wilcoxon test between the most relevant methods identified by the Friedman test and the percentage of certainty of statistical difference.

According to Table 9, FCTAN shows the best performance regarding the Friedman test for both horizons. Additionally, the percentage of certainty that there is a statistical difference among the methods is 99.4% and 100%
Table 6: Forecast errors results of sMAPE average for all individual methods.

| Subset | Horizon | AA | ET | NN | TB | St | RW | TH | NA | AR | BA | SP | HO | FF | JA | SM | PA | FI | PE | BAC | BAB | FCTA |
|--------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|------|
| Y      | 6       | 6.67 | 6.84 | 12.6 | 6.65 | 18.0 | 7.17 | 6.85 | 7.12 | 8.46 | 5.61 | 5.52 | 6.91 | 6.45 | 6.70 | **6.11** | 6.68 | 6.79 | 6.79 | 8.39 | 3.49 |
| Q      | 8       | **4.32** | 4.72 | 6.45 | 4.38 | 51.4 | 4.71 | 4.80 | 5.27 | 12.0 | 4.75 | 5.97 | 4.61 | 4.48 | 4.45 | 4.29 | 4.40 | 4.33 | **4.28** | 4.54 | 3.59 |
| M      | 18      | 7.24 | 8.18 | 8.43 | 7.19 | 19.7 | 8.31 | 7.69 | 8.31 | 10.2 | 7.93 | 17.9 | 8.63 | 7.17 | 7.23 | 6.37 | **6.32** | 6.38 | 6.38 | 15.2 | 13.5 |
| W      | 13      | 5.97 | 5.89 | 7.88 | 5.56 | 29.1 | 6.35 | 5.91 | 6.20 | 10.0 | 5.93 | 8.94 | 6.26 | **5.41** | 5.61 | 5.98 | 5.52 | 5.65 | **5.48** | 7.53 | 4.43 |
| D      | 14      | 3.66 | 3.47 | 3.85 | 3.56 | 11.6 | 4.03 | 3.43 | **3.39** | 4.62 | 3.64 | 5.19 | 3.56 | 3.51 | 3.52 | 3.46 | **3.13** | 3.48 | 3.48 | 3.67 | 3.57 |

*Standard Normal distribution (N), Uniform distribution (U).
for horizon 1 (H1) and n (HN), respectively. Therefore, for H1, we executed a Wilcoxon test comparing FCTAN with BICN; this test shows that FCTAN has statistically the same performance as BICN. Furthermore, the Wilcoxon test for HN compared the results of FCTAN with BICN, showing a statistical difference with a percentage of certainty of 99.8%.

The sum of negative ranges is larger than the sum of positive ranges; the sMAPE error of the FCTA method is smaller than the BICN method. On the other hand, the p-value equals 0.198 (see Table 12), showing that both methods have the same statistical performance. Table 11 shows the Wilcoxon test results applied to FCTAN and
BICN forecast methods. The sum of negative ranges is larger than the sum of positive ranges. Therefore, the sMAPE error of the FCTAN method is smaller than the BICN method. Finally, the p-value equals 0.002 (see Table 12), indicating that both methods have a statistically significant difference.

Thus, according to the Wilcoxon tests, there is no significant difference between FCTAN and BICN in H1.

In Hn FCTAN outperforms the other methods. Finally, to verify the quality of the FCTA, we use the metric average [2], but we use sMAPE with equation (5). Figure 15 shows
Table 9: Friedman and Wilcoxon for H1 and Hn horizon results.

| Horizon | Friedman Methods | Wilcoxon Method with Winner | % Certainty |
|---------|-----------------|-----------------------------|-------------|
| 1       | (1) FCTAN       | FCTAN – BICN                | 99.4        |
|         | (2) BICN       |                             |             |
|         | (3) FCTAU      |                             |             |
|         | (4) BICU       |                             |             |
| n       | (1) FCTAN      | FCTAN-BICN                  | 99.8        |
|         | (2) FCTAU      |                             |             |
|         | (3) BICN       |                             |             |
|         | (4) BICU       |                             |             |

The results of Wilcoxon for H1 and Hn horizon can be verified in Tables 10–12. Table 10 shows the Wilcoxon test for the FCTAN and BICN.

Table 10: Wilcoxon’s W-test with ranges to the H1-horizon.

| FCTAN – BICN | N   | Mean range | Sum of ranges |
|--------------|-----|-------------|---------------|
| Negative ranges | 60a | 54.07       | 3244.00       |
| Positive ranges | 46b | 52.76       | 2427.00       |
| Tie           | 44c |             |               |
| Total         | 150 |             |               |

*FCTAN < BICN.  9FCTAN > BICN.  *FCTAN = BICN.

Table 11: Wilcoxon’s W-test with ranges to the Hn-horizon.

| FCTAN–BICN | N   | Mean range | Sum of ranges |
|------------|-----|-------------|---------------|
| Negative ranges | 71a | 53.45       | 3795.00       |
| Positive ranges | 35b | 53.60       | 1876.00       |
| Tie         | 44c |             |               |
| Total       | 150 |             |               |

*FCTAN < BICN.  9FCTAN > BICN.  *FCTAN = BICN.

Table 12: Wilcoxon’s W statistical test results for H1 and Hn subsets.

| Subset | Methods | Z     | p value |
|--------|---------|-------|---------|
| H1     | FCTAN–BICN | −1.29 | 0.198   |
| Hn     | FCTAN–BICN | −3.02 | 0.002   |

Figure 15: Error ratio of the proposed FCTAN compared to other ensemble prediction algorithms.
the error ratio of the best-proposed combination methodology FCTAN compared to other ensemble prediction algorithms; according to this metric, FCTAN has the best performance than all the other methodologies of the comparison.

\[
\text{Error ratio} = \frac{s\text{MAPE}_{\text{evaluating algorithm}}}{s\text{MAPE}_{\text{baseline algorithm}}}
\]  

(5)

6. Conclusion

In the forecasting area, enormous defiance are related to improving the ability to obtain good forecasts in various scenarios where the data sets are coming from diverse areas to obtain good predictions for \( n \) periods ahead in the horizon. This work presents the FCTA forecasting methodology for time series from different scenarios, which can forecast \( n \)-periods-ahead.

The FCTA architecture and sequence diagram are included for supporting future applications.

The proposed methodology, as in previous works, assigns weights to the individual forecasting methods in the ensemble, FCTA, has two implementation modes: (A) initial ponderation, and the enhanced stage are both performed with FCTA techniques and (B) initial ponderation comes from another methodology, but it is enhanced with FCTA techniques. The first mode has produced FCTAU and FCTAN. The second mode produced BICU and BICN. These methods were tested with datasets taken from M4-competition, and hypothesis test were performed.

FCTAU and FCTAN were compared in a first experiment with the best reference methods of M4 and using almost twenty thousand time series from several areas. In one-period-ahead, and \( n \)-period-ahead, FCTAN surpassed all these methods. Besides, in a second experiment, FCTA methods were compared with those of the state-of-the-art: LA, and PP-MA. The experimentation for one period, and \( n \)-periods-ahead, shows that FCTAN obtains the best results followed by FCTAU.

The analysis of the results for the methods produced by FCTA in the second mode, BICU, and BICN, for one-period-ahead show the following ranking: FCTAN, BICN, FCTAU, BICU. For \( n \)-periods-ahead, the ranking is FCTAN, FCTAU, BICN, and BICU. Therefore, BIC hybridized with FCTA obtained best results than the state-of-the-art methods. Consequently, FCTA is a good methodology and can be hybridized with other methods with the condition that they can produce good rankings of individual forecasting methods. Thus, FCTA is a significant contribution to the ensemble forecasting area.

Data Availability

All data generated or analyzed during this study are included in this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to acknowledge CONACYT (Consejo Nacional de Ciencia y Tecnología), TecnNM/Instituto Tecnológico de Ciudad Madero, and the National Laboratory of Information Technologies (LaNTI) for access to the cluster.

References

[1] S. Tofighy, A. A. Rahmanian, and M. Ghobaei-Arani, “An ensemble CPU load prediction algorithm using a Bayesian information criterion and smooth filters in a cloud computing environment,” Software: Practice and Experience, vol. 48, no. 12, pp. 2257–2277, 2018.

[2] A. A. Rahmanian, M. Ghobaei-Arani, and S. Tofighy, “A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment,” Future Generation Computer Systems, vol. 79, pp. 54–71, 2018.

[3] S. F. Crone, M. Hibon, and K. Nikolopoulos, “Advances in forecasting with neural networks? Empirical evidence from the NNS competition on time series prediction,” International Journal of Forecasting, vol. 27, no. 3, pp. 635–660, 2011.

[4] J. J. Ruiz-Aguilar, I. Turias, J. González-Enrique, D. Uría, and D. Elizondo, “A permutation entropy-based EMD-ANN forecasting ensemble approach for wind speed prediction,” Neural Computing & Applications, vol. 33, no. 7, pp. 2369–2391, 2020.

[5] A. N. Khan, N. Iqbal, R. Ahmad, and D.-H. Kim, “Ensemble prediction approach based on learning to statistical model for efficient building energy consumption management,” Symmetry, vol. 13, no. 3, pp. 1–26, 2021.

[6] T. McAndrew, N. Wattanachit, G. C. Gibson, and N. G. Reich, “Aggregating predictions from experts: a review of statistical methods, experiments, and applications,” WIREs Computational Statistics, vol. 13, no. 2, pp. 1–26, 2020.

[7] M. Etemadi, M. Ghobaei-Arani, and A. Shahidinejad, “Resource provisioning for IoT services in the fog computing environment: an autonomic approach,” Computer Communications, vol. 161, pp. 109–131, 2020.

[8] R. Prudêncio and T. Ludermir, “Using machine learning techniques to combine forecasting methods,” in Lecture Notes in Computer Science, vol. 3339, pp. 1122–1127, 2004.

[9] C. Lemke and B. Gabrys, “Meta-learning for time series forecasting and forecast combination,” Neurocomputing, vol. 73, no. 10–12, pp. 2006–2016, 2010.

[10] R. B. C. Prudêncio and T. B. Ludermir, “Meta-learning approaches to selecting time series models,” Neurocomputing, vol. 61, no. 1–4, pp. 121–137, 2004.

[11] M. Kuck, S. F. Crone, and M. Freitag, “Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data,” in Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, June 2016.

[12] Y. Kang, R. J. Hyndman, and K. Smith-Miles, “Visualising forecasting algorithm performance using time series instance spaces,” International Journal of Forecasting, vol. 33, no. 2, pp. 345–358, 2017.

[13] P. Montero-Manso, G. Athanasopoulos, R. J. Hyndman, and T. S. Talagala, “FFORMA: Feature-Based Forecast Model Averaging,” International Journal of Forecasting, vol. 36, 2020 http://business.monash.edu/econometrics-and-business-statistics/research/publications.

[14] C. K. Chan, B. G. Kingsman, and H. Wong, “Determining when to update the weights in combined forecasts for product
demand—an application of the CUSUM technique,” European Journal of Operational Research, vol. 153, no. 3, pp. 757–768, 2004.

[15] X. Chen, Y. Jiang, K. Yu, Y. Liao, J. Xie, and Q. Wu, “Combined time-varying forecast based on the proper scoring approach for wind power generation,” Journal of Engineering, vol. 2017, no. 14, pp. 2655–2659, 2017.

[16] J. M. Bates and C. W. J. Granger, “The combination of forecasts,” Journal of the Operational Research Society, vol. 20, no. 4, pp. 451–468, 1969.

[17] R. S. Tsay, G. Dueck and T. Scheuer, “qQthreshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing,” Journal of Computational Physics, vol. 90, no. 1, pp. 161–175, 1990.

[18] R. T. Clemen, “Combining forecasts: a review and annotated bibliography,” International Journal of Forecasting, vol. 5, no. 4, pp. 559–583, 1989.

[19] J. S. Armstrong, “Combining forecasts,” in Principles of Forecasting: A Handbook for Researchers and Practitioners, pp. 417–439, Springer, Boston, MA, 1st ed. edition, 2001.

[20] A. C. B. Mancuso and L. Werner, “Review of combining forecasts approaches,” Independent Journal of Management & Production, vol. 4, no. 1, pp. 248–277, 2013.

[21] G. Dueck and T. Scheuer, “Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing,” Journal of Computational Physics, vol. 90, no. 1, pp. 679–688, 2006.

[22] P. Goodwin and R. Lawton, “On the asymmetry of the symmetric MAPE,” International Journal of Forecasting, vol. 15, no. 4, pp. 405–408, 1999.

[23] P. Newbold and C. W. J. Granger, “Experience with forecasting univariate time series and the combination of forecasts,” Journal of the Royal Statistical Society: Series A, vol. 137, no. 2, pp. 131–165, 1974. http://www.jstor.org/stable/23444536.

[24] R. L. Winkler and S. Makridakis, “The combination of forecasts,” Journal of the Royal Statistical Society: Series A, vol. 146, no. 2, pp. 150–157, 1983.

[25] M. Deutsch, C. W. J. Granger, and T. Terasvirta, “The combination of forecasts using changing weights,” International Journal of Forecasting, vol. 10, no. 1, pp. 47–57, 1994.

[26] R. Adhikari and R. K. Agrawal, “Performance evaluation of weights selection schemes for linear combination of multiple forecasts,” Artificial Intelligence Review, vol. 42, no. 4, pp. 529–548, 2014.

[27] C. T. West, “System-based weights versus series-specific weights in the combination of forecasts,” Journal of Forecasting, vol. 15, no. 5, pp. 369–383, 1996.

[28] C. W. J. Granger and R. Ramanathan, “Improved methods of combining forecasts,” Forecast, vol. 3, pp. 197–204, 1984.

[29] S. Crone, “Time Series Forecasting Competition for Computational Intelligence,” 2008, http://www.neural-forecasting-competition.com.

[30] S. Jaganathan and P. K. S. Prakash, “A combination-based forecasting method for the M4-competition,” International Journal of Forecasting, vol. 36, no. 1, pp. 98–104, 2020.