Thailand Statistician
October 2024; 22(4): 963-985
http://statassoc.or.th
Contributed paper

Enhancing Decomposition and Holt-Winters Weekly Forecasting of PM2.5 Concentrations in Thailand's Eight Northern Provinces Using the Cuckoo Search Algorithm

Watha Minsan* [a], Pradthana Minsan [b] and Wararit Panichkitkosolkul [c]

[a] Data Science Research Center, Department of Statistics, Faculty of Science, Chiang Mai University, Chiang Mai, Thailand.

[b] Department of Mathematics and Statistics, Faculty of Science and Technology, Chiang Mai Rajabhat University, Chiang Mai, Thailand.

[c] Department of Mathematics and Statistics, Thammasat University, Pathumthani, Thailand.

*Corresponding author; e-mail: wathaminsan@gmail.com

Received: 23 February 2024
Revised: 9 June 2024
Accepted: 9 July 2024

Abstract

This research aims to introduce hybrid models that integrate the Cuckoo Search Algorithm with Holt-Winters (CS-HW) and Decomposition (CS-D) for time series forecasting of weekly PM2.5 concentrations in Thailand's eight northern provinces. The study consists of two phases: the training dataset phase and the testing dataset phase. During the training dataset phase, the Cuckoo Search (CS) algorithm demonstrates effective parameter optimization capabilities, seamlessly integrating with Holt-Winters and decomposition models. This integration results in lower Root Mean Square Error (RMSE) values compared to classical approaches, including Grid Search for Holt-Winters (Classic-HW) and Classical Decomposition (Classic-D). In the testing dataset phase, key performance metrics such as RMSE, Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are utilized. The results indicate that the CS-HW and CS-D models outperform other methods in weekly forecasting of PM2.5 concentrations across several provinces, including Chiang Mai, Chiang Rai, Lamphun, Lampang, Mae Hong Son, and Phayao. Notably, the Box-Jenkins model outperformed other methods in Nan, while in Phrae, the Long Short-Term Memory (LSTM) model demonstrates other forecasting performance.

Keywords: Cuckoo Search algorithm, decomposition, Holt-Winters, PM2.5, time series analysis.

1. Introduction

Thailand's economic growth, transitioning from an agriculture-based to a more industrialized economy, has led to a significant increase in air pollution. This shift has fueled urbanization, characterized by increased vehicle traffic, industrial activities, and construction, all contributing to higher levels of air pollution throughout the country.

The upper northern region of Thailand faces particularly severe air pollution challenges due to several factors. The region's mountainous geography creates a valley effect, especially around Chiang Mai. These mountains act like a bowl, trapping pollutants within the valley, especially during the dry season from November to April when wind is less prevalent to disperse particles. Seasonal activities, such as burning agricultural residues and forest fires, further elevate PM2.5 levels during this period. Additionally, temperature inversions can trap pollutants closer to the ground, worsening air quality. Cross-border pollution from neighboring countries like Myanmar and Laos also contributes to the PM2.5 concentration in Northern Thailand (Pollution Control Department 2019).

High levels of PM2.5 are associated with various health issues, including respiratory problems like asthma and bronchitis, as well as more severe conditions such as lung cancer. These fine particles can also exacerbate cardiovascular diseases and affect cognitive functions. In response, government agencies and the private sector have established air quality monitoring stations across the region to monitor PM2.5 levels in real-time. This monitoring is crucial for issuing timely warnings and enabling people to take preventative measures based on the data provided by these agencies. Raising public awareness about the risks posed by PM2.5 is essential. Campaigns focus on educating the public about the health dangers and promoting preventive actions, including wearing masks and reducing outdoor activities during high pollution periods.

Forecasting PM2.5 time series is critically important due to the presence of both trend and seasonality in the data, presenting a challenging task. Accurately forecasting PM2.5 is crucial for several reasons. Understanding long-term trends in PM2.5 data, influenced by factors such as urbanization, industrial growth, and changes in emission regulations, is vital for precise long-term forecasting. Seasonal variations in PM2.5 levels, often significant and driven by changes in weather conditions, agricultural practices, and atmospheric conditions, are essential for improving short-term forecasting accuracy.

Decomposition and Holt-Winters (HW) smoothing techniques are powerful tools in statistical research, enabling experts to uncover patterns in sequential data and enhance the precision of future observations. Introduced by Persons (1919), decomposition techniques have evolved over time and continue to be instrumental in forecasting. Brown's pioneering work in 1959 (Brown 1959) contributed significantly to the development and refinement of exponential smoothing methods, including HW, which remains widely used today. In the realm of air quality research, Pozza et al. (2010) investigated PM2.5 and PM10 concentrations in Sao Carlos, Brazil, while Ao et al. (2019) studied PM2.5 concentrations in Hefei, China. Both studies utilized the HW smoothing techniques for prediction, comparing their results with the Box-Jenkins method. Expanding the scope to long-term forecasting, Nath et al. (2021) analyzed forecasts related to PM2.5 and PM10 in Kolkata, India, employing methods such as HW, Box-Jenkins, and deep learning methodologies. In a different approach, Zhao et al. (2022) used decomposition methods to examine PM2.5 in Beijing, China. They subsequently proposed a forecasting model based on hybrid ARIMA, integrating the Akaike Information Criterion (AIC) and the advanced grid search fixed-order methods with seasonal decomposition.

In this research, we concentrate on integrating Classical Decomposition (Classic-D) and HW methods, augmented by optimization metaheuristics, to achieve precise parameter estimation in statistical forecasting models. Previous studies have demonstrated the effectiveness of hybrid approaches combining HW with optimization metaheuristics to enhance parameter estimation efficiency. For instance, Assis et al. (2013) combined HW with the ant colony optimization algorithm, a technique originally developed by Dorigo (1992) and further refined by Dorigo and Stützle (2004). Eusébio et al. (2015) applied HW in forecasting scenarios with double seasonality, utilizing a range of optimization algorithms including Hill Climber, Simulated Annealing (introduced by Kirkpatrick et al. 1983), genetic algorithms (introduced by Holland 1975), and Particle Swarm Optimization (introduced by Kennedy and Eberhart 1995). Furthermore, Jiang et al. (2020) integrated HW with the fruit fly optimization method, introduced by Pan (2011, 2012). Azmi (2013) focused on parameter estimation for the HW method using genetic algorithms. Similarly, Simoni et al. (2015) employed particle swarm optimization, an evolutionary algorithm, alongside HW for optimization in hydropower plants. Minsan and Minsan (2023, 2024) introduced an innovative approach by incorporating the decomposition model and the HW model into the whale optimization algorithm. Lastly, Mauricio and Ostia (2023) effectively utilized the Cuckoo Search algorithm (CS) to enhance the HW method in distribution transformer load forecasting.

The CS is a metaheuristic algorithm that is inspired by the brood parasitism behavior of cuckoo birds. Some cuckoo birds lay their eggs in the nests of other birds, known as host birds. Host birds may either discard or abandon the nest if they detect the foreign eggs. To avoid this, cuckoo birds have developed various strategies to camouflage their eggs so that the host birds mistake them for their own. Introduced by Xin-She Yang and Suash Deb in 2009, the CS algorithm is inspired by the cuckoo bird's strategies to find optimal solutions (Yang and Deb 2009). This study introduces a model that integrates Cuckoo Search algorithm with Holt-Winters smoothing (CS-HW) and Decomposition techniques (CS-D) for forecasting the weekly PM2.5 concentration levels across eight provinces in the northern region of Thailand. The CS algorithm proves pivotal in generating multiple parameters essential for both the HW and decomposition methods. Notably, while the HW model requires estimation of three parameters, the decomposition method is more intricate, requiring the estimation of up to 54 parameters. We compare the forecasting results obtained from the CS-HW and CS-D approaches against those sourced from the Classic-D method and the grid search for HW model (Classic-HW). In summary, our model aims to significantly enhance the accuracy of forecasting weekly PM2.5 concentration levels for a two-year period in advance in eight northern provinces of Thailand.

2. Research Methods

2.1. Cuckoo Search (CS) algorithm

The CS is a nature-inspired optimization technique developed by Yang and Deb (2009). It has been effectively applied to a wide range of optimization problems, from engineering design to machine learning applications. The effectiveness of CS lies in its simplicity and its capability to avoid local optima, making it a highly versatile tool for complex optimization tasks. To facilitate understanding, the standard CS algorithm can be described based on three fundamental rules:

- Each cuckoo lays one egg in a nest chosen at random. For simplicity, it is assumed that the number of eggs, nests, and cuckoos is the same.
- Nests containing the highest quality eggs (representing the best solutions) are retained for the next generation.
- Given the limited number of nests, there is a discovery probability $p_a \in (0,1)$ associated with each cuckoo’s egg. This probability determines whether the host bird will eject the cuckoo’s egg or abandon the nest and build a new one.

The CS algorithm utilizes the concept of Lévy flights to simulate the searching behavior of cuckoos through a random walk process. Lévy flights are a specific type of random walk characterized by step lengths that follow heavy-tailed probabilistic distributions. This mechanism of Lévy flights enables the algorithm to perform localized random walks, interspersed with longer jumps. Such a strategy is effective in escaping local optima and allows for a more comprehensive exploration of the search space.

In mathematical terms, the movement vector X_i^{t+1} of a cuckoo i is modeled using Lévy flights, which facilitate a global random walk. This approach is detailed in Yang (2014):

$$X_i^{t+1} = X_i^t + \alpha \text{Lévy}(s, \lambda) Z, \tag{1}$$

where X_i^t is a cuckoo i in iteration t, $\text{Lévy}(s, \lambda) = s\left(X_i^t - X^*\right)$ and $Z \sim N(0,1)$.

The current best solution is X^*. In Mantegna's algorithm, the step size s is computed using two Gaussian distributions, U and V, via the following transformation, as described by Mantegna (1994):

$$s = \frac{U}{|V|^{1/\lambda}},$$

where

$$U \sim N(0, \sigma^2), V \sim N(0,1).$$

Here, U represents samples drawn from a Gaussian normal distribution with a mean of zero and a variance of σ^2. The variance can be calculated by

$$\sigma^2 = \left(\frac{\Gamma(1+\lambda)}{\lambda \Gamma((1+\lambda)/2)} \cdot \frac{\sin(\pi \lambda/2)}{2^{(\lambda-1)/2}} \right)^{1/\lambda}.$$

Yang (2014) established the Lévy parameters as $\lambda = 1.5$ and step-size scaling factor as $\alpha = 0.01$ for most problems. Consequently, σ^2 is determined to be 0.6965745.

The CS involves a process where host birds may abandon their nests (discard eggs) with a fraction $p_a \in (0,1)$, also referred to as the assigned probability, to create entirely new nests. Initially, a random number $p \in (0,1)$ is chosen; if $p < p_a$, then the vector X_i^{t+1} is selected and modified, otherwise, it remains unchanged. The local random walk is described as follows:

$$X_i^{t+1} = X_i^t + Z(X_j^t - X_k^t), \tag{2}$$

where X_j^t and X_k^t are two different solutions selected randomly by random permutation. And Yang (2014) suggested $n = 15$ to 40 (by default value is 25) and $p_a = 0.25$ are sufficient for most optimization problem. The steps of the CS are succinctly outlined in the pseudo-code presented in Figure 1.

2.2. Data preparation

The dataset for this study was sourced from the Pollution Control Department (2023) of Thailand's Ministry of Natural Resources and Environment. It is publicly available through their official website, Air4Thai. Our study concentrates on daily PM2.5 concentration data collected from eight provinces in the northern region of Thailand, namely Chiang Mai (CMI) station code 36T, Lampang (LPG) station code 37T, Chiang Rai (CRI) station code 57T, Mae Hong Son (MSN) station code 58T, Nan (NAN) station code 67T, Lamphun (LPN) station code 68T, Phrae (PRE) station code 69T, and Phayao (PYO) station code 70T, as coded by The Royal Gazette (Thailand) (2021). The data spans from January 1, 2019, to June 30, 2023, and has been aggregated to calculate weekly averages of PM2.5 concentrations, resulting in a total of 235 weeks. The dataset was divided into two datasets. The first dataset, designated as the training dataset, consists of 130 weeks and was used for developing models across various forecasting methodologies. The second dataset, labeled the test dataset, includes 105 weeks. This division was deliberately structured to cover the two-year forecast period targeted in our study, aligning with the objective of providing long-term forecasts useful for informed policy decisions.

```
The number of bird host nests: $n$ , the number of parameters: $d$ , the maximum number of iterations: $T_{\max}$,
the time limit: $MaxTime$ , and the fitness value fails to improve after a specified number of iterations: $T_{improve}$.
                              Cuckoo Search Via Lévy flights
-------------------------------------------------------------------------------------------------------
Objective function $f(X), X = (x_1,...,x_d)^T$ .
Generate initial population of $n$ host nests $X_i$ ( $i = 1,2,...,n$) .
While (t< $T_{\max}$ ) or (time < $MaxTime$ ) or (the fitness value fails to improve after a specified < $T_{improve}$ )
    For $i$ =1 to $n$
        Get a cuckoo $i$ by performing Lévy flights equation (1).
        Calculate fitness of a cuckoo $i$ .
        Update a host nest $X_i$ if there is a better solution.
    End for
    For $i$ =1 to $n$
        Random $p$
        If ( $p < p_a$ ),
            A worse nest is abandoned and new one is built equation (2).
            Calculate fitness of a new nest $i$ .
            Update a host nest $X_i$ if there is a better solution.
        End if
    End for
    Rank the solutions and find the current best $X^*$ , update $t = t + 1$ .
End while
Return $X^*$ .
```

Figure 1 Pseudo-code of the CS

2.3. The components of a time series

For effective model selection in time series forecasting, understanding the series' components is vital. This study employs statistical tests and in-depth analysis of the time series plots, along with an examination of data characteristics, to gain a comprehensive understanding of these components.

Understanding the components of a time series is crucial for the selection of an appropriate forecasting model. In this study, statistical tests and thorough examinations of the time series plots are employed alongside an analysis of the inherent data characteristics to facilitate a comprehensive understanding of these components.

2.3.1 The runs test

This statistical method is utilized to detect the presence of a trend in time series data. It evaluates whether consecutive values in the series consistently show an increasing or decreasing pattern. To minimize seasonal influences, the time series data is initially processed through a central 52-week moving average. This process, referred to as deseasonalization (Y^T), aids in a more transparent analysis of the underlying trend. After deseasonalization, the Runs Test is applied to the altered dataset for a deeper investigation of its trend characteristics.

2.3.2 The Kruskal-Wallis test

This non-parametric statistical test is employed to assess significant differences across groups. Commonly used for comparing multiple independent groups, it can also be adapted to analyze seasonal variations in time series data. To isolate the seasonal component, the trend must be removed. This is accomplished by calculating $Y^S = Y - Y^T$, where Y^S is the detrended data. In this equation, Y^T is the trend component removed from the original time series Y. The detrended data, Y^S is subsequently used in the Kruskal-Wallis test to explore seasonal effects.

2.3.3 Autocorrelation Function (ACF)

Detecting Trends: In a time series that shows a trend, the ACF typically displays a gradual decline in correlation coefficients as the lag increases. This pattern emerges because observations closer in time (shorter lags) are more strongly correlated in the presence of a trend, as they are influenced by similar long-term progressions. In contrast, observations at greater distances apart (longer lags) tend to exhibit weaker correlations. This gradual reduction in the ACF is indicative of a series with a trend.

Uncovering Seasonality: When seasonality is present, particularly in time series data with weekly patterns, the ACF will display periodic spikes at specific lags. These lags correspond to the seasonal cycle, which, for weekly data, is typically a year. Therefore, in a time series with a distinct weekly seasonal pattern, the ACF would show significant peaks at lags of 52 weeks, 104 weeks, and so on. These peaks signify the presence of a recurring pattern that occurs at regular intervals each year, which is a hallmark of seasonality in the dataset.

2.3.4 Levene’s test

Levene’s Test is a statistical method used to evaluate homoscedasticity across different groups. While it is traditionally employed to compare variances among independent samples, this test can also be adapted to analyze homoscedasticity within time series data. This adaptation is crucial for determining whether the variability in the time series is consistent over time, which is an essential factor in effective time series forecasting.

2.4. Forecasting model

2.4.1 Classical Decomposition (Classic-D) method

The classical decomposition method is an approach to time series forecasting that involves dissecting a time series into its distinct components, namely trend, seasonal, and residual elements. The typical steps in classical decomposition forecasting include:

1. Data preparation: This step involves collecting and organizing a sufficient historical dataset for forecasting purposes.

2. Visualization: The data is visualized using time series plots to identify underlying patterns, trends, and seasonality.

3. Seasonal period identification: This involves determining the duration of recurring cycles within the data.

4. Detrending: The trend component is removed to focus on seasonality and residuals. This is typically done using centered moving averages. In an additive model, the centered moving average values are subtracted from the original time series. In a multiplicative model, they are divided by the original time series.

5. Seasonality estimation: Average values for each season are calculated and then adjusted to determine the seasonal component. In an additive model, this involves subtracting the overall average from each seasonal value. In a multiplicative model, it involves dividing each seasonal value by the overall average.

6. Deseasonalization: A deseasonalized series is obtained by either subtracting (in an additive model) or dividing (in a multiplicative model) the seasonal component from the original time series.

7. Trend calculation: Linear regression is used to identify and quantify the trend component.

8. Forecasting: Future values are predicted by combining the forecasted trend and seasonal components. Equations (3)-(6) are used for both analytical modeling and prospective forecasting,

Additive modeling: $$Y_t = \beta_0 + \beta_1 t + S_t + \varepsilon_t, \tag{3}$$

Additive forecasting: $$\hat{Y}_t = \hat{\beta}_0 + \hat{\beta}_1 t + \hat{S}_t, \tag{4}$$

Multiplicative modeling: $$Y_t = (\beta_0 + \beta_1 t) \times S_t \times \varepsilon_t, \tag{5}$$

Multiplicative forecasting: $$\hat{Y}_t = (\hat{\beta}_0 + \hat{\beta}_1 t) \times \hat{S}_t, \tag{6}$$

where Y_t is the observed data at time t. $\hat{Y}_t$ is the forecasted data at time t. ε_t is the residual at time t. t is the time index. β_0 and β_1 are, respectively, the y-intercept and the slope coefficient. $\hat{\beta}_0$ and $\hat{\beta}_1$ are the estimated coefficients of β_0 and β_1. S_t is seasonal component at time t, which belongs to a specific season i ($i = 1,...,s$). We define s as a 52-week cycle. Each time point t thus falls into one of these 52 distinct seasons. $\hat{S}_t$ is estimated seasonal component of S_t.

9. Evaluation and refinement: This step involves assessing the forecast against the actual data to gauge its accuracy and effectiveness. In this study, we utilized the commercial software Minitab, as it follows these steps.

2.4.2 Holt-Winters (Classic-HW) method

The HW method is adept at handling both additive and multiplicative seasonal fluctuations within time series data. For the additive model, the specific computations are detailed in equations (7) through (10). In contrast, the calculations for the multiplicative model are outlined in equations (11) through (14). This methodological distinction allows for accurate modelling of different types of seasonal variations.

Additive forecasting: $$\hat{Y}_{t+p} = \hat{T}_t + p\hat{\beta}_t + \hat{S}_{t-s+1+((p-1)\bmod s)} \quad \text{for} \quad p = 1, 2, ... \quad , \tag{7}$$

$$\hat{T}_t = \alpha(Y_t - \hat{S}_{t-s}) + (1-\alpha)(\hat{T}_{t-1} + \hat{\beta}_{t-1}), \tag{8}$$

$$\hat{\beta}_t = \gamma(\hat{T}_t - \hat{T}_{t-1}) + (1-\gamma)\hat{\beta}_{t-1}, \tag{9}$$

$$\hat{S}_t = \delta(Y_t - \hat{T}_t) + (1-\delta)\hat{S}_{t-s}, \tag{10}$$

Multiplicative forecasting: $$\hat{Y}_{t+p} = (\hat{T}_t + p\hat{\beta}_t) \times \hat{S}_{t-s+1+((p-1)\bmod s)} \quad \text{for} \quad p = 1, 2, ..., \tag{11}$$

$$\hat{T}_t = \alpha(Y_t / \hat{S}_{t-s}) + (1-\alpha)(\hat{T}_{t-1} + \hat{\beta}_{t-1}), \tag{12}$$

$$\hat{\beta}_t = \gamma(\hat{T}_t - \hat{T}_{t-1}) + (1-\gamma)\hat{\beta}_{t-1}, \tag{13}$$

$$\hat{S}_t = \delta(Y_t / \hat{T}_t) + (1-\delta)\hat{S}_{t-s}, \tag{14}$$

where p is the number of time periods ahead you want to forecast, s set at 52 weeks in this study, $\hat{T}_t$ is the level of the time series, $,\hat{\beta}_t$ is the trend, and $\hat{S}_t$ is the seasonality component.

In this study, the smoothing coefficients $\alpha,\ \gamma,$ and δ are crucial and must be within the range of 0 to 1. These coefficients significantly impact how the model updates its level, trend, and seasonal components, balancing the influence of current observations and previously smoothed values. A lower coefficient, closer to 0, leads to more pronounced smoothing, emphasizing historical data trends. Conversely, values near 1 give more weight to recent observations, resulting in less smoothing. In some cases, as seen in commercial programs like Minitab, smoothing coefficients can be set to the extremes of 0 or 1. Coefficients at 1 lead to the least smoothed, or virtually unsmoothed, version of the original time series. In contrast, coefficients at 0 result in the smoothest representation of the pattern that the actual time series follows, as discussed by Montgomery et al. (2007).

To optimize these parameters, we utilize the grid search method. This method entails systematically varying the coefficients in increments of 0.01, covering a range from 0 to 1. Such an exhaustive search entails a total of $101^3 = 1{,}030{,}301$ iterations. Our approach, referred to as the Classic-HW method, identifies the optimal parameters by minimizing the RMSE. For this study, we utilized Python Colab, as it efficiently facilitates these steps.

$$\text{Objective Minimize RMSE}(\alpha,\gamma,\delta),$$

$$\text{Variable range } \begin{cases} 0 \le \alpha \le 1 \\ 0 \le \gamma \le 1, \\ 0 \le \delta \le 1 \end{cases}$$

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{t=1}^{m}\left(Y_t - \hat{Y}_t\right)^2}\ , \tag{15}$$

where m is the length of dataset, with $m = 130$ for the training phase and $m = 235$ for the future forecasting; Y_t is the actual value, and $\hat{Y}_t$ is the forecasted value produced by Classic-HW.

2.4.3 Hybrid of the Cuckoo Search algorithm with Holt-Winters (CS-HW)

The CS algorithm is employed to optimize the $\alpha, \gamma,$ and δ parameters for the HW model. The computational steps for this process are detailed in the pseudo-code presented in Figure 2.

The performance of the HW model, featuring the optimized parameters, is evaluated through forecast accuracy assessment. This involves comparing the forecasted data against the actual dataset. The objective function for the CS-HW model is outlined in equation (15), where $\hat{Y}_t$ is the forecasted value produced by CS-HW. For this study, we utilized Python Colab, as it efficiently facilitates these steps.

The number of bird host nests: $n = 25$, the number of parameters: $d = 3$, the maximum number of iterations: $T_{\max} = 1{,}000$, the time limit: $MaxTime = 30$ sec., and the fitness value fails to improve after a specified number of iterations: $T_{improve} = 300$.

```
Cuckoo Search Via Lévy flights
----------------------------------------------------------------
Objective function $f(X), X = (x_1, x_2, x_3)^T$ .
Generate initial population of $n$ host nests $X_i$ ( $i = 1, 2, ..., n$) .
While (t< $T_{\max}$ ) or (time < $MaxTime$ ) or (the fitness value fails to improve after a specified < $T_{improve}$ )
    For $i$ = 1 to $n$
        Get a cuckoo $i$ by performing Lévy flights (1).
        Calculate fitness using HW by (15) of a cuckoo $i$.
        Update a host nest $X_i$ if there is a better solution.
    End for
    For $i$ = 1 to $n$
        Random $p$
        If ( $p < p_a$ ),
            A worse nest is abandoned and new one is built (2).
            Calculate fitness using HW by (15) of a new nest $i$ .
            Update a host nest $X_i$ if there is a better solution.
        End if
    End for
    Rank the solutions and find the current best $X^*$ , update $t = t + 1$ .
End while
Return $X^*$
# $x_1 = \alpha, x_2 = \gamma, x_3 = \delta$
# Objective Minimize RMSE($\alpha^*, \gamma^*, \delta^*$) where $\alpha^*, \gamma^*, \delta^*$ are the optimized parameters.
```

Figure 2 Pseudo-code of the CS-HW

2.4.4 Hybrid of the Cuckoo Search algorithm with Decomposition (CS-D)

The CS algorithm is employed to optimize the parameters of the decomposition model, specifically $\hat{\beta}_0, \hat{\beta}_1$ and $\hat{S}_1, \hat{S}_2, ..., \hat{S}_s$. The aim of the study is to demonstrate the efficiency of hybridizing CS with decomposition, particularly when optimizing a large number of parameters. This process is outlined in the pseudo-code provided in Figure 3.

Scaling Parameters

The scaling parameters in Figure 3, introduced by Minsan and Minsan (2023, 2024), follow these steps:

1) Setting constraints for upper and lower bounds of parameters:

1.1) Constraints on the upper and lower bounds of $\hat{\beta}_0$ and $\hat{\beta}_1$.

Calculate the trend component using linear regression on the dataset to obtain $\hat{\beta}_0'$, and $\hat{\beta}_1'$. It is recommended to constrain the upper and lower bounds of the parameters $\hat{\beta}_0$, and $\hat{\beta}_1$ according to the following equation:

Constraint the upper bound of $\hat{\beta}_0$ as $\hat{\beta}_0' + 0.2\left|\hat{\beta}_0'\right|$ and $\hat{\beta}_1$ as $\hat{\beta}_1' + 0.2\left|\hat{\beta}_1'\right|$,

Constraint the lower bound of $\hat{\beta}_0$ as $\hat{\beta}_0' - 0.2\left|\hat{\beta}_0'\right|$ and $\hat{\beta}_1$ as $\hat{\beta}_1' - 0.2\left|\hat{\beta}_1'\right|$.

1.2) Constraints on the upper and lower bounds of $\hat{S}_i$ ($i = 1, 2, ..., s$).

To address seasonal variations, the time series is initially detrended using first-order differencing $\Delta Y_t = Y_t - Y_{t-1}$. The establishment of upper and lower constraints for the seasonal parameters $\hat{S}_1, \hat{S}_2, ..., \hat{S}_s$ are proposed as follows:

Constraint the upper bound (UB) is +[extreme value of amplitude of ΔY_t],

Constraint the lower bound (LB) is –[extreme value of amplitude of ΔY_t].

2) The CS is configured to search for parameters within the boundary of [0, 1]. Consequently, adjusting the units of the parameters before calculating the fitness value becomes essential. The equation employed for this purpose is as follows:

Original Value = Scaled Value × (Constraint the upper bound – Constraint the lower bound) + Constraint the lower bound. (16)

After applying (16), if the original values are denoted as $\hat{S}_i$, the seasonal adjustment is calculated using the following formula:

Additive decomposition: Adjust $\hat{S}_i = \hat{S}_i - \sum_{i=1}^{s} \frac{\hat{S}_i}{s}$, then $\sum_{i=1}^{s} \hat{S}_i = 0$.

Multiplicative decomposition: Adjust $\hat{S}_i = \dfrac{\hat{S}_i}{\sum_{i=1}^{s} \frac{\hat{S}_i}{s}}$, then $\sum_{i=1}^{s} \hat{S}_i = s$.

In this equation, the original value refers to the parameter value in its original data unit, while the scaled value is the one obtained by the CS algorithm within the range of [0, 1]. This step is particularly crucial when dealing with parameters of varying units and a large number of parameters. The objective function of CS-D to the following equation:

Objective Minimize RMSE($\hat{\beta}_0, \hat{\beta}_1, \hat{S}_1, ..., \hat{S}_s$),

$$\text{Variable range} \begin{cases} \hat{\beta}_0' - 0.2\left|\hat{\beta}_0'\right| \le \hat{\beta}_0 \le \hat{\beta}_0' + 0.2\left|\hat{\beta}_0'\right| \\ \hat{\beta}_1' - 0.2\left|\hat{\beta}_1'\right| \le \hat{\beta}_1 \le \hat{\beta}_1' + 0.2\left|\hat{\beta}_1'\right| \\ \text{LS} \le \hat{S}_i \le \text{US) for } i = 1, 2, ..., s \end{cases},$$

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{t=1}^{m}\left(Y_t - \hat{Y}_t\right)^2}, \tag{17}$$

where m is the length of dataset, with $m = 130$ for the training phase and $m = 235$ for the future forecasting; Y_t is the actual value, $s = 52$, and $\hat{Y}_t$ is the forecasted value produced by CS-D.

For this study, we utilized Python Colab, as it efficiently facilitates these steps.

The number of bird host nests: $n = 25,$ the number of parameters: $d = 54,$ the maximum number of iterations: $T_{\max} = 1{,}000,$ the time limit: $MaxTime = 30$ sec., and the fitness value fails to improve after a specified number of iterations: $T_{improve} = 300.$

Cuckoo Search Via Lévy flights

```
Objective function $f(X), X = (x_1, x_2, ..., x_{54})^T$.
Generate initial population of $n$ host nests $X_i$ ($i = 1, 2, ..., n$).
While (t< $T_{\max}$) or (time < $MaxTime$) or (the fitness value fails to improve after a specified < $T_{improve}$)
    For $i$ = 1 to $n$
        Get a cuckoo $i$ by performing Lévy flights (1).
        Scaling Parameters
        Calculate fitness using decomposition by the equation (17) of a cuckoo $i$.
        Update a host nest $X_i$ if there is a better solution.
    End for
    For $i$ = 1 to $n$
        Random $p$
        If ( $p < p_a$ ),
            A worse nest is abandoned and new one is built (2).
            Scaling Parameters
            Calculate fitness using decomposition by the equation (17) of a new nest $i$.
            Update a host nest $X_i$ if there is a better solution.
        End if
    End for
    Rank the solutions and find the current best $X^*$, update $t = t + 1$.
End while
Return $X^*$
# $x_1 = \hat{\beta}_0, x_2 = \hat{\beta}_1, x_3 = \hat{S}_1, ..., x_{54} = \hat{S}_{52}$
# Objective Minimize RMSE($\hat{\beta}_0^*, \hat{\beta}_1^*, \hat{S}_1^*, \hat{S}_2^*, ..., \hat{S}_{52}^*$) where $\hat{\beta}_0^*, \hat{\beta}_1^*, \hat{S}_1^*, \hat{S}_2^*, ..., \hat{S}_{52}^*$ are the optimized parameters.
```

Figure 3 Pseudo-code of the CS-D

2.4.5 Box-Jenkins method

The Box-Jenkins methodology is a systematic approach to identifying, fitting, and checking Seasonal Autoregressive Integrated Moving Average (SARIMA) models for time series data. The process involves several key steps:

1) Data preparation

Unit root detection: Use the Dickey-Fuller (DF) test to detect unit roots and assess the stationarity of the time series data. The maximum lag order is set at 52 to account for seasonality.

Model selection: Apply the Akaike Information Criterion (AIC) to facilitate model selection by minimizing its value.

Transforming non-stationary data: Transform non-stationary time series data to stationary forms through differencing or Box-Cox transformation (natural logarithm transformation). Differencing is used to obtain the d and D values.

2) Model identification

ACF and Partial Autocorrelation Function (PACF): Examine the ACF and PACF plots of the stationary data to identify the order of the autoregressive (AR) and moving average (MA) components.

Grid search for hyperparameter tuning: Employ a grid search, an optimization technique in machine learning, to tune the hyperparameters p and q (bounded between 0 and 2) and the seasonal components P and Q (bounded between 0 and 1). This systematic search helps in selecting the best combination of parameters that minimize the AIC value.

3) Model estimation

Parameter estimation: Estimate the parameters of the identified SARIMA model using methods such as Maximum Likelihood Estimation (MLE).

Model fitting: Fit the SARIMA model to the time series data using the estimated parameters.

4) Model validation

Residual analysis: The process of selecting the order with the lowest AIC continues until all tests are passed for the model residuals. These tests include:

- Normality assessment: Use the Kolmogorov-Smirnov test (KS) to assess the normality of the residuals.
- Zero mean verification: Verify that the mean of the residuals is zero using the t-test.
- Constant variance check: Ensure constant variance of the residuals with Levene’s test.
- Independence check: Check for statistical independence of the residuals from other time points using the Ljung-Box test.

The model that meets these criteria and has the minimum AIC value is chosen as the forecasting model.

5) Forecasting

Generate forecasts: Use the fitted SARIMA model to generate forecasts for future time periods. The forecasts are based on the AR MA SAR and SMA components identified and the differencing applied.

2.4.6 Long Short-Term Memory (LSTM) method

LSTM is a type of recurrent neural network (RNN) particularly effective for modeling sequential data and capturing long-term dependencies. LSTM networks are especially suitable for time series forecasting due to their ability to retain information over extended periods. The process for applying LSTM to time series forecasting involves several key steps:

1) Data preparation

Data Scaling: Normalize the PM2.5 concentration data to a range suitable for neural networks (typically between 0 and 1) to ensure efficient training and convergence.

Training and Testing Split: Divide the dataset into training and testing subsets. The training data is used to train the LSTM model, while the testing data is used to evaluate its performance.

Sequence Generation: Create input-output sequences from the time series data. For example, if the lookback period is 52 weeks, the input sequence will consist of 52 consecutive weeks, and the corresponding output will be the PM2.5 concentration of the next week.

2) Model configuration

Network Architecture: Define the architecture of the LSTM network, including the number of layers and the number of neurons in each layer. Common configurations include one or more LSTM layers followed by dense layers.

Hyperparameter tuning: Experiment with different hyperparameter settings to identify the optimal configuration. Key hyperparameters include:

- Loss function: Mean Square Error (MSE) is typically used as the loss function for regression tasks.
- Optimizer: The Adam optimizer is commonly used for training LSTM networks due to its efficiency and adaptability.
- Epochs: Set the number of epochs (iterations over the entire training dataset) to 100.
    - Training Duration and Learning Stability: Setting the epoch count to 100 ensures that the model has enough iterations to learn the underlying patterns in the time series data. This duration allows the model to converge to an optimal solution, minimizing the loss function effectively.
    - Avoiding Overfitting: An epoch count of 100 strikes a balance where the model is sufficiently trained without overfitting. This number was chosen based on empirical evidence from preliminary experiments indicating the model starts to overfit if trained for significantly more than 100 epochs.
- Look back: Experiment with different lookback periods, such as 26, 52, and 104 weeks.
    - Seasonal Patterns: The selected lookback values (26, 52, 104) correspond to weekly data, capturing key seasonal patterns. Specifically, 26 weeks cover half a year, 52 weeks cover a full year, and 104 weeks cover two years. This choice ensures that the model captures both short-term and long-term seasonal variations. These values were chosen based on preliminary experiments that demonstrated their effectiveness in capturing significant patterns in the data.
- Number of Neurons in Layers: Test various configurations, including LSTM layers with 25, 50, 75, and 100 neurons.
    - Model complexity: The chosen neuron counts (25, 50, 75, 100) were selected to explore varying levels of model complexity. Lower counts like 25 and 50 allow the model to learn simpler patterns, while higher counts like 75 and 100 enable the model to capture more complex relationships in the data.
    - Performance Optimization: Preliminary testing showed that these neuron counts provided a good balance between model performance and computational efficiency, allowing the model to capture essential patterns without overfitting.

3) Model training

Training process: Train the LSTM model using the training dataset. The training process involves forward propagation, where the input sequences are passed through the network to obtain predictions, and backpropagation, where the errors are propagated back through the network to update the weights.

Validation: Use a validation subset of the training data to monitor the model's performance and prevent overfitting.

4) Model evaluation

Evaluation metrics: Assess the performance of the trained LSTM model using key metrics such as RMSE, MAE, and MAPE.

5) Forecasting

Generate forecasts: Use the trained LSTM model to generate forecasts for future time periods. The forecasts are based on the learned patterns and dependencies in the training data.

Inverse transformation: Apply the inverse of the normalization step to convert the forecasts back to the original scale of the PM2.5 concentration data. The implementation was carried out in Python.

2.5. Assessment metrics

The evaluation framework is divided into two distinct categories:

The first category aims to identify the optimal modeling approach by assessing which yields the lowest RMSE during the training dataset phase. The primary objective during this phase is to optimize the model parameters to achieve the best fit. This involves fine-tuning the variable values to minimize the objective function, which is inherently tied to the training dataset. This process involves comparing the RMSE of various models, specifically Classic-D with CS-D and Classic-HW with CS-HW, to determine the most effective approach. The RMSE is calculated using the following equation:

$$\text{RMSE} = \sqrt{\frac{1}{n_1}\sum_{t=1}^{n_1}\left(Y_t - \hat{Y}_t\right)^2}\ ,$$

where $n_1 = 130,$ which is the length of the train dataset; Y_t and $\hat{Y}_t$ are the actual value and forecasting value of the train dataset, respectively.

The second category evaluates the model’s forecasting capabilities over a two-year or 105-week period. To comprehensively compare model performance, three key metrics are employed: RMSE, MAE, and MAPE. Each of these metrics provides unique insights into the model's accuracy:

- RMSE helps identify the model's overall error magnitude.
- MAE provides a straightforward measure of the average error, which is less sensitive to outliers than RMSE.
- MAPE offers a perspective on the error in percentage terms, making it easier to interpret the model’s accuracy relative to the actual values.

Using these three metrics allows for a more comprehensive evaluation of the models' performance in the testing dataset phase, ensuring that different aspects of accuracy and robustness are captured. Their respective formulas are as follows:

$$\text{RMSE} = \sqrt{\frac{1}{105}\sum_{t=131}^{n_2}\left(Y_t - \hat{Y}_t\right)^2},$$

$$\text{MAE} = \frac{1}{105}\sum_{t=131}^{n_2}\left|Y_t - \hat{Y}_t\right|,$$

and

$$\text{MAPE} = \frac{100}{105}\sum_{t=131}^{n_2}\left|\left(Y_t - \hat{Y}_t\right)/Y_t\right|,$$

where $n_2 = 235,$ which is the length of the test dataset; Y_t and $\hat{Y}_t$ are the actual value and forecasting value of the test dataset, respectively.

3. Results and Discussion

The trend, seasonal, and variance characteristics of the data were evaluated using the runs test, Kruskal-Wallis test, and Levene’s test, as detailed in Table 1. Further analysis of the time series plots and the inherent characteristics of the data provided additional insights. The runs test indicated a trend in all provinces, evidenced by p-values less than the predetermined significance level. The Kruskal-Wallis test also revealed seasonal variations in all provinces, as the p-values were below the significance threshold. Additionally, the Autocorrelation Function (ACF), depicted in Figures 4 and 5, reinforced these findings, supporting the use of this dataset in our experiments. Levene’s test results for Chiang Mai, Lampang, and Phrae suggested non-constant variance, with P-values lower than the established significance level. Thus, for Chiang Mai, Lampang, and Phrae, the data appear to exhibit multiplicative seasonal characteristics. In contrast, for the remaining provinces, where p-values did not fall below the significance level, the data seem to exhibit additive seasonal characteristics.

Table 1 The test statistics and p-values from runs, Kruskal-Wallis, and Levene's tests for each of the provinces

		CMI	LPG	CRI	MSN	NAN	LPN	PRE	PYO
Runs	z	-12.636	-12.339	-12.636	-12.934	-13.231	-12.934	-12.636	-12.934
	p-value	0.000*	0.000*	0.000*	0.000*	0.000*	0.000*	0.000*	0.000*
Kruskal Wallis	H	143.48	133.45	136.97	138.48	138.57	133.29	147.56	137.3
	p-value	0.000*	0.000*	0.000*	0.000*	0.000*	0.000*	0.000*	0.000*
Levene's	Levene	3.93	5.95	2.1	3.87	0.8	0.11	5.5	1.73
	p-value	0.049*	0.015*	0.1487	0.0502	0.3719	0.7441	0.020*	0.1899

Note: * denotes that the null hypothesis is rejected at the 0.05 significance level.

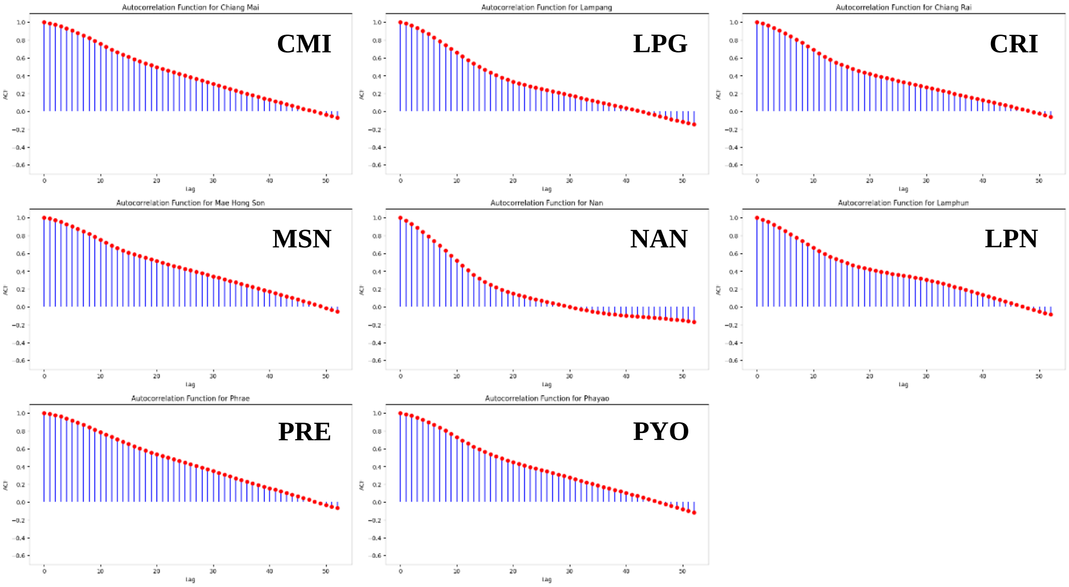

Figure 4 ACF of the deseasonalization time series data of 8 northern provinces

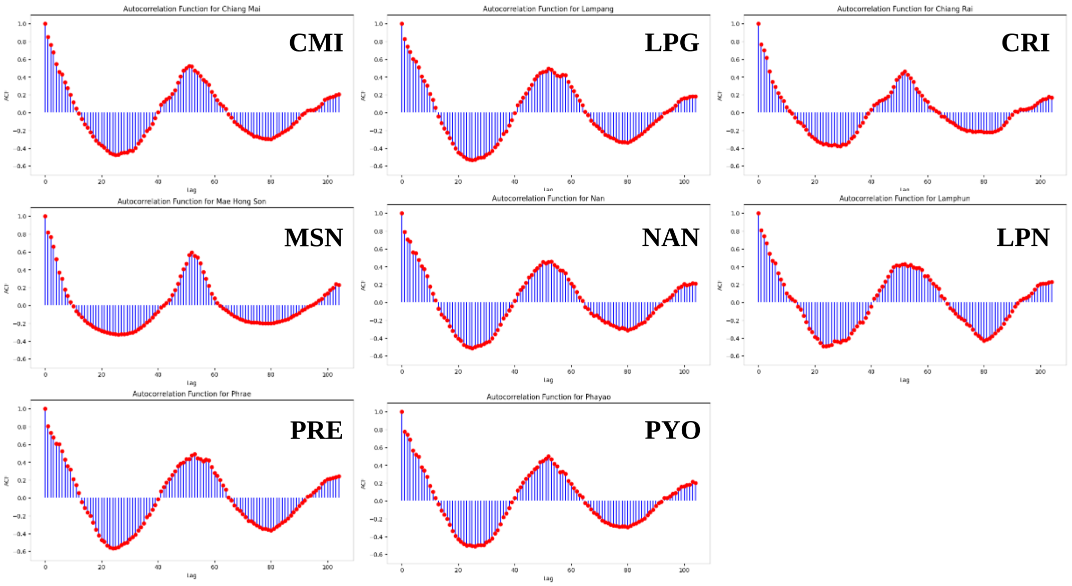

Figure 5 ACF of the detrended time series data of 8 northern provinces

3.1. Comparative performance analysis in the train dataset phase

This section evaluates the performance of the CS-HW and CS-D models in forecasting PM2.5 concentrations by conducting a comparative analysis with two classical forecasting models: Classic-D and Classic-HW. When applied to the same dataset, these models produced varying results, as detailed in Table 2.

Table 2 RMSE of train dataset for each province

	CMI	LPG	CRI	MSN	NAN	LPN	PRE	PYO
CS-D	9.4372	10.8857	13.5602	15.5523	9.7764	9.3755	10.4844	10.9220
Classic-D	11.1428	14.6171	17.1819	20.4583	12.5993	12.4362	13.7693	15.0596
CS-HW	13.9743	17.8788	21.4752	26.3050	16.0405	15.1030	18.2299	16.7295
Classic-HW	14.0183	17.8802	21.4755	26.3059	16.0411	15.1055	18.2300	16.7296

Note: The lowest RMSE value between CS-D and Classic-D, and between CS-HW and Classic-HW is highlighted

The CS algorithms demonstrated effective parameter optimization capabilities, enhancing both the HW and decomposition models. This hybridization resulted in lower RMSE values compared to classical approaches. Specifically, in pairwise comparisons between CS-HW and Classic-HW, a marginal yet notable improvement in RMSE was observed for CS-HW. This improvement is attributed to the compatibility of grid search techniques with the HW model, despite the computational cost being a potential limitation. Notably, the grid search involved 1,030,301 computational loops, indicating extensive computation time. In contrast, the CS-D model displayed a markedly lower RMSE compared to Classic-D. This finding suggests that the CS algorithm’s extensive parameter optimization capabilities substantially outperform traditional methods. In summary, hybridizing CS algorithms with HW and Decomposition methods facilitates more precise model fitting than classical approaches.

3.2. Comparative performance analysis in the test dataset phase

In this study, long-term forecasts spanning a two-year period (or 105 weeks) were generated. To assess the performance of various forecasting models, key metrics such as the RMSE, MAE, and MAPE were employed. Two traditional statistical models, Classic-HW and Classic-D, were compared against advanced models hybridizing the Cuckoo Search algorithm, namely CS-HW and CS-D. Additionally, the study included the Box-Jenkins SARIMA model (Ao 2019, Nath 2021, and Box et al. 1994) and the Long Short-Term Memory (LSTM) technique (Nath 2021, and Zaini et al. 2022) for comprehensive comparative analysis.

A summary of model performance metrics across various provinces is presented in Tables 3 (RMSE), 4 (MAE), and 5 (MAPE). Notably, the CS-D model exhibited the lowest MAE and MAPE values in both Chiang Mai and Lampang, and had the lowest RMSE and MAE values in Mae Hong Son and Phayao. Therefore, it emerged as the most effective model for these provinces. In Nan, the Box-Jenkins model with SARIMA$(2,1,0)(0,1,1)_{52}$, which benefited from a Box-Cox transformation using natural logarithm for the data series, emerged as the top performer. It met assumptions of residual normal distribution, constant variance (homoscedasticity), independence, and a mean of zero, and achieved the lowest RMSE and MAE. In both Chiang Rai and Lamphun, CS-HW had the best RMSE and MAE, while in Phrae, the LSTM model outperformed others in RMSE, MAE, and MAPE.

For the overall two-year forecast period spanning from July 1, 2023, to June 30, 2025, covering 104 weeks, specific forecasting methods were assigned to the entire dataset (n). Based on various metrics, CS-D model was used for forecasting in four provinces, CS-HW in two provinces, Box-Jenkins in one province, and LSTM in one province.

Table 3 RMSE of test dataset for each province

	CMI	LPG	CRI	MSN	NAN	LPN	PRE	PYO
CS-D	22.2009	21.6636	29.1500	24.3623	24.7608	21.1321	18.5221	22.3587
CS-HW	30.1382	21.4035	28.1133	27.7118	23.8437	16.2253	16.1647	23.3082
Classic-D	21.8527	24.1968	31.1981	30.6131	25.2392	21.7873	19.2807	24.9190
Classic-HW	24.4562	21.3767	28.1254	27.7224	23.8242	16.2925	16.1709	23.3159
Box-Jenkin	24.0092	24.2997	35.3172	28.8855	22.5499	17.4173	16.8644	23.5297
LSTM	23.5010	22.7592	34.5332	34.9457	25.4916	19.6430	15.8903	22.8028

Note: The lowest RMSE value for each province is highlighted

Table 4 MAE of test dataset for each province

	CMI	LPG	CRI	MSN	NAN	LPN	PRE	PYO
CS-D	11.7627	12.7736	17.2695	12.8361	18.1644	13.8320	11.0377	12.4824
CS-HW	16.9473	13.0339	15.6056	13.4583	15.2811	10.6339	10.4521	12.6385
Classic-D	12.1583	14.6528	19.0095	15.1466	18.4372	13.9921	12.3533	15.1587
Classic-HW	12.8128	13.0197	15.6057	13.3378	15.2759	10.7055	10.4510	12.6433
Box-Jenkin	12.8686	15.1068	21.8719	13.4957	13.5623	11.4518	10.4930	13.1559
LSTM	12.1720	14.5197	17.9776	22.6446	13.9615	11.3641	9.4822	15.2884

Note: The lowest MAE value for each province is highlighted

Table 5 MAPE of test dataset for each province

	CMI	LPG	CRI	MSN	NAN	LPN	PRE	PYO
CS-D	39.9733	37.7203	73.0373	55.5084	58.8285	54.9558	35.0065	43.9017
CS-HW	55.5875	40.2646	65.3643	45.0804	44.5349	51.6014	39.2250	42.7957
Classic-D	40.5221	44.1965	78.3786	57.6796	61.2800	54.3404	43.6607	56.3023
Classic-HW	40.1776	40.2480	65.1668	43.0756	44.5342	52.0895	39.1758	42.7776
Box-Jenkin	43.2071	46.6772	97.2940	39.8519	39.3229	52.2111	39.5831	42.4107
LSTM	46.7234	46.3732	64.7125	107.1493	36.0099	46.6055	32.3585	60.7292

Note: The lowest MAPE value for each province is highlighted

4. Conclusions

When utilizing the entire time series dataset (n) to implement the forecasting methods for a two-year period from July 1, 2023, to June 30, 2025, encompassing 104 weeks, the forecasts are visually represented in Figure 6. A close examination of the visual data for each province reveals that both actual and forecasted PM2.5 concentration values effectively capture the seasonal patterns across all provinces. The forecasting trends vary depending on the methodology employed.

The CS-D forecasting method exhibits forecasts that closely align with the variability of the actual data, showing slight upward or downward trends, depending on the province under consideration. In contrast, the CS-HW forecasts demonstrate a smoothing of the actual data's variability through a more gradual adjustment. The Box-Jenkins forecasts display distinct upward trends with clarity for Nan province. Lastly, the LSTM forecasts yield the smoothest values, with hyperparameters optimized from experiments at a lookback of 52 and a layer count of 50.

This study successfully forecasts PM2.5 concentrations in the eight northern provinces of Thailand, regions that have consistently experienced high PM2.5 concentrations from the end of winter to the middle of the summer season for several years. Notably, the forecasts reveal elevated PM2.5 concentrations from weeks 5 to 18, coinciding with the end of winter through the middle of the summer season, as detailed in Tables 6 to 8.

In Tables 6 to 8, the severity of PM2.5 concentrations is clearly marked. The yellow category, which indicates health impacts, ranges from 37.5 to 75.09 micrograms per cubic meter. Above this, the red category signals severe health impacts, starting at 75.1 micrograms per cubic meter. These forecasts highlight the critical periods when PM2.5 concentrations reach levels that could significantly affect public health. As such, they serve as a crucial reminder for authorities and the public to implement and adhere to preventive measures. Strengthening air quality management and enhancing public awareness campaigns during these peak periods could mitigate the adverse health effects associated with high PM2.5 levels.

In conclusion, this research introduces a novel forecasting method that hybridizes the cuckoo search algorithm with the Holt-Winters and decomposition models. This approach has demonstrated enhanced predictive performance compared to traditional models in our experiments across all eight provinces. Future research aims to not only refine these computational techniques but also to extend their application to other time series datasets, thus evaluating their adaptability and effectiveness in a broader range of scenarios.

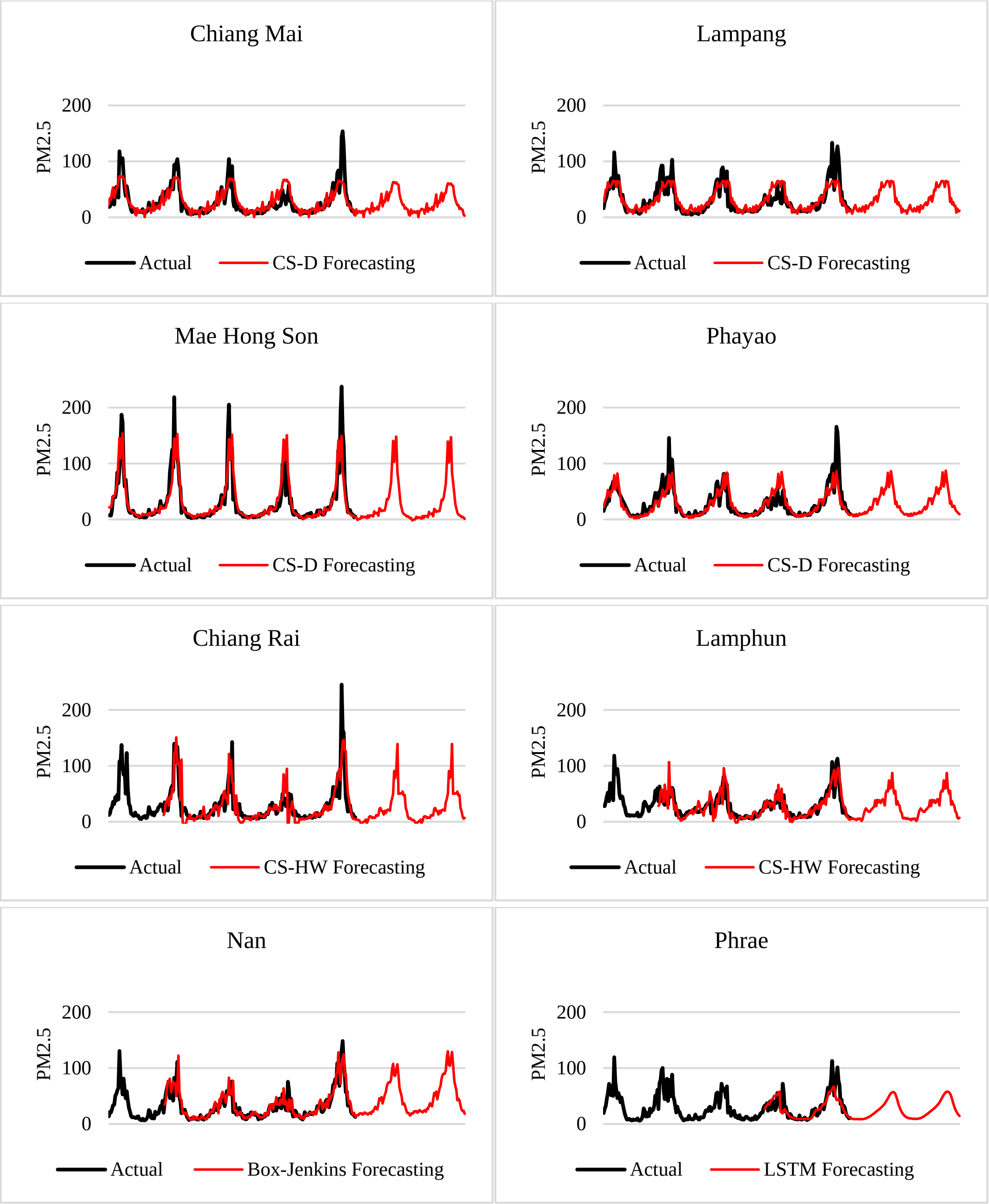

Figure 6 Actual and forecasted values for each province using the most suitable forecasting method

Table 6 Forecasted values for all provinces in the year 2023

		Chiang		Chiang	Mae Hong				
Month Year	Weekly	Mai	Lampang	Rai	Son	Nan	Lamphun	Phrae	Phayao
July-23	27	14	12	4	4	15	5	10	9
July-23	28	7	6	3	1	16	7	9	9
July-23	29	10	12	3	0	16	5	9	6
July-23	30	11	18	2	1	19	5	9	8
July-23	31	8	22	1	2	18	4	9	6
August-23	32	9	15	1	6	19	3	9	10
August-23	33	11	11	1	4	17	6	9	6
August-23	34	0	10	1	5	18	4	9	7
August-23	35	11	16	1	4	21	6	9	11
September-23	36	12	12	4	5	19	5	9	9
September-23	37	16	11	1	2	19	1	9	9
September-23	38	15	19	1	7	17	5	9	11
September-23	39	14	18	8	3	19	13	9	10
October-23	40	7	9	10	8	20	20	9	14
October-23	41	16	21	9	6	18	24	10	11
October-23	42	26	16	6	8	23	22	11	12
October-23	43	10	16	8	5	22	19	12	16
October-23	44	17	21	7	15	27	18	13	18
November-23	45	14	23	7	11	31	18	14	22
November-23	46	15	27	12	11	27	22	15	17
November-23	47	19	21	11	12	26	23	16	19
November-23	48	13	25	11	7	36	26	18	28
December-23	49	22	35	23	20	46	27	19	37
December-23	50	29	32	25	19	42	38	21	34
December-23	51	42	38	19	15	48	28	22	37
December-23	52	19	28	19	16	37	39	24	27

Note: The yellow category, which has health impacts, ranges from 37.5 to 75.09 micrograms per cubic meter. The red category, indicating severe health impacts, starts at 75.1 micrograms per cubic meter and above.

Table 7 Forecasted values for all provinces in the year 2024

		Chiang		Chiang	Mae Hong				
Month Year	Weekly	Mai	Lampang	Rai	Son	Nan	Lamphun	Phrae	Phayao
January-24	1	29	37	13	16	45	35	25	34
January-24	2	27	49	20	16	52	39	27	40
January-24	3	39	48	15	25	57	33	28	45
January-24	4	46	63	20	37	68	36	30	57
January-24	5	30	55	21	35	73	40	32	50
February-24	6	36	54	21	41	75	36	34	45
February-24	7	45	57	20	52	74	29	37	55
February-24	8	45	61	35	65	79	51	39	54
February-24	9	56	65	41	101	98	56	43	66
March-24	10	63	65	50	141	108	54	46	84
March-24	11	63	63	91	103	87	74	49	58
March-24	12	61	54	78	131	86	59	53	67
March-24	13	62	65	91	148	98	61	55	87
April-24	14	59	64	139	85	107	87	57	71
April-24	15	58	64	50	68	88	59	57	57
April-24	16	41	51	51	48	64	50	56	44
April-24	17	33	27	50	29	56	55	54	27
April-24	18	27	34	51	21	50	31	48	32
May-24	19	22	23	54	13	35	37	42	22
May-24	20	23	27	47	10	37	31	35	22
May-24	21	15	20	48	9	35	29	30	24
May-24	22	17	21	24	8	27	19	26	17
June-24	23	15	8	19	5	21	16	22	14
June-24	24	14	15	9	6	19	7	19	13
June-24	25	4	10	5	4	19	6	16	10
June-24	26	3	12	7	2	15	7	14	9
July-24	27	14	12	4	3	17	5	13	9
July-24	28	7	6	3	1	20	7	12	10

Table 7 (Continued)

Month Year	Weekly	Chiang Mai	Lampang	Chiang Rai	Mae Hong Son	Nan	Lamphun	Phrae	Phayao
July-24	29	9	12	3	1	19	5	11	6
July-24	30	11	18	2	1	23	5	10	9
July-24	31	8	22	1	0	22	4	10	6
August-24	32	9	15	1	5	23	3	10	11
August-24	33	11	11	1	2	20	6	9	7
August-24	34	0	10	1	4	22	4	9	8
August-24	35	11	16	1	3	25	6	9	12
September-24	36	12	12	4	4	23	5	9	9
September-24	37	15	11	1	1	22	1	9	10
September-24	38	14	19	1	6	21	5	9	12
September-24	39	13	18	8	2	23	13	10	11
September-24	40	7	9	10	6	25	20	10	14
October-24	41	16	21	9	5	22	24	11	12
October-24	42	25	16	6	7	27	22	11	13
October-24	43	10	16	8	4	26	19	12	16
October-24	44	16	21	7	14	32	18	13	19
November-24	45	13	23	7	10	37	18	15	23
November-24	46	14	27	12	10	33	22	16	18
November-24	47	18	21	11	11	31	23	17	20
November-24	48	12	25	11	6	44	26	19	29
December-24	49	21	35	23	19	55	27	20	38
December-24	50	28	32	25	17	51	38	22	35
December-24	51	41	38	19	14	58	28	23	38
December-24	52	19	28	19	15	44	39	25	27
December-24/ January-25	53/1	28	37	13	15	54	35	27	35

Note: The yellow category, which has health impacts, ranges from 37.5 to 75.09 micrograms per cubic meter. The red category, indicating severe health impacts, starts at 75.1 micrograms per cubic meter and above.

Table 8 Forecasted values for all provinces in the year 2025

Month Year	Weekly	Chiang Mai	Lampang	Chiang Rai	Mae Hong Son	Nan	Lamphun	Phrae	Phayao
January-25	2	26	49	20	15	63	39	28	41
January-25	3	37	48	15	24	68	33	30	45
January-25	4	44	63	20	35	82	36	32	58
January-25	5	29	55	21	34	88	40	34	51
February-25	6	35	54	21	40	90	36	36	45
February-25	7	43	57	20	51	89	29	39	56
February-25	8	44	61	35	64	96	51	42	55
February-25	9	54	65	41	100	118	56	45	67
March-25	10	60	65	50	139	130	54	49	84
March-25	11	60	64	91	102	105	74	52	59
March-25	12	59	54	78	130	104	59	55	67
March-25	13	60	65	91	147	118	61	57	87
March-25	14	57	64	139	84	129	87	58	72
April-25	15	56	64	50	66	106	59	58	58
April-25	16	40	52	51	47	78	50	57	44
April-25	17	32	27	50	28	68	55	54	28
April-25	18	26	34	51	19	60	31	50	33
May-25	19	21	23	54	12	42	37	44	23
May-25	20	22	27	47	9	45	31	37	23
May-25	21	15	20	48	8	42	29	31	25
May-25	22	17	21	24	6	33	19	26	18
June-25	23	14	8	19	4	26	16	22	15
June-25	24	13	15	9	5	23	7	19	13
June-25	25	4	10	5	3	23	6	16	11
June-25	26	3	12	7	1	18	7	14	9

Note: The yellow category, which has health impacts, ranges from 37.5 to 75.09 micrograms per cubic meter. The red category, indicating severe health impacts, starts at 75.1 micrograms per cubic meter and above.

Acknowledgments

This study is supported by the Data Science Research Center, Department of Statistics, Faculty of Science, Chiang Mai University, Thailand.

References

Ao X, Yuan H, Zhang D. PM2.5 analysis and prediction based on seasonal time series model. IOP Conf. Series: Earth and Environmental Science 371 [monograph online]. 2019 [cited 2023 Jul 30]; 052006: 1-9. Available from: doi:10.1088/1755-1315/371/5/052006.

Assis MVO, Carvalho LF, Rodrigues JJPC, Proença ML. Holt-Winters statistical forecasting and ACO metaheuristic for traffic characterization. Proceeding of 2013 IEEE International Conference on Communications (ICC); 2013; Budapest; Hungary. p. 2524-2528.

Azmi NILBM. Parameters estimation of Holt-Winter smoothing method using genetic algorithm. Master [thesis] [monograph online]. Malaysia: Universiti Teknologi; 2013 [cited 2023 Sep 25]. Available from: https://eprints.utm.my/32356/1/NurIntanLiyanaMohdAzmiMFS2013.pdf.

Box GEP, Jenkins GM, Reinsel GC. Time series analysis: forecasting and control, 3rd ed. New Jersey: Prentice Hall; 1994.

Brown RG. Statistical forecasting for inventory control. New York: McGraw-Hill; 1959.

Dorigo M. Optimization, learning and natural algorithms. PhD [thesis]. Italy: Politecnico di Milano; 1992.

Dorigo M, Stützle T. Ant colony optimization. London: The MIT Press; 2004.

Eusébio E, Camus C, Curvelo C. Metaheuristic approach to the Holt-Winters optimal short term load forecast. Renewable Energy and Power Quality Journal. 2015; 1(13): 708-713.

Holland JH. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Michigan: University of Michigan Press; 1975.

Jiang W, Wu X, Gong Y, Yu W, Zhong X. Holt–Winters smoothing enhanced by fruit fly optimization algorithm to forecast monthly electricity consumption. Energy [serial on the internet]. 2020 [cited 2023 Sep 1]; 193: 116779. Available from: doi.org/10.1016/ j.energy.2019.116779.

Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks. 1995; 4: 1942-1948.

Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science. 1983; 220(4598): 671-680.

Mantegna RN. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic process. Phys Rev E. 1994: 49(5): 4677-4683.

Mauricio CC, Ostia CF. Cuckoo search algorithm optimization of Holt-Winter method for distribution transformer load forecasting. Proceeding of the 2023 9th International Conference on Control, Automation and Robotics (ICCAR) Beijing; China. 2023. p. 36-42.

Minsan W, Minsan P, Incorporating decomposition and the Holt-Winters method into the whale optimization algorithm for forecasting monthly government revenue in Thailand. Science & Technology Asia [serial on the internet]. 2023 [cited 2024 Feb 1]; 28(4): 38-53. Available from: https://ph02.tci-thaijo.org/index.php/SciTechAsia/article/view/250335.

Minsan W, Minsan P, Decomposition and Holt-Winters enhanced by the whale optimization algorithm for forecasting the amount of water inflow into the large dam reservoirs in southern Thailand. Journal of Current Science and Technology [serial on the internet]. 2024 [cited 2024 Jun 1]; 14(2): Article 38. Available from: https://doi.org/10.59796/jcst.V14N2.2024.38.

Montgomery DC, Jennings CL, Kulahci M. Introduction time series analysis and forecasting. New Jersey: John Wiley & Sons; 2007.

Nath P, Saha P, Middya AI, Roy S. Long-term time-series pollution forecast using statistical and deep learning methods. Neural Computing and Applications. 2021; 33: 12551-12570.

Pan WT. A new evolutionary computation approach: Fruit fly optimization algorithm. Proceedings of the Conference of Digital Technology and Innovation Management; Taipei; Taiwan. 2011. p. 382-391.

Pan WT. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. Knowledge-Based Systems [serial on the internet]. 2012 [cited 2023 Sep 1]. 26: 69-74. Available from: https://doi.org/10.1016/j.knosys.2011.07.001.

Persons WM. II. the method used. Review of Economics and Statistics [monograph online]. 1919 [cited 2023 Jul 30]; 1(2): 117-139. Available from: http://www.jstor.org/stable/1928600.

Pollution Control Department. Archived Data (Daily). Ministry of Natural Resources and Environment [monograph online]. 2023 [cited 2023 Jul 29]. Available from: http://air4thai.pcd.go.th. (in Thai)

Pollution Control Department. Standard Operating Procedure for Northern Haze Response. Ministry of Natural Resources and Environment [serial on the internet]. 2019 [cited 2023 Jul 29]. Available from: http://air4thai.com/tagoV2/tago_file/books/book_file/b2b90b38b2b1026ed2c29b001e17d05f.pdf. (in Thai)

Pozza SA, Lima EP, Comin TT, Gimenes M, Coury J. Time series analysis of PM2.5 and PM10-2.5 mass concentration in the city of Sao Carlos, Brazil. International Journal of Environment and Pollution. 2010; 41(1-2): 90-108.

Simoni A, Gjika ED, Puka L. Evolutionary algorithm PSO and Holt Winters method applied in hydro power plants optimization. Proceeding of the Conference: SPNA-Statistics Probability and Numerical Analysis [monograph online]. 2015 [cited 2023 Jul 20]. p. 7-20. Available from: https://www.researchgate.net/publication/311581451_EVOLUTIONARY_ALGORITHM_PSO_AND_HOLT_WINTERS_METHOD_APPLIED_IN_HYDRO_POWER_PLANTS_OPTIMIZATION.

The Royal Gazette (Thailand). Ratchakitcha, The Secretariat of the Cabinet [monograph online]. 2021 [cited 2023 Aug 6]. Available from: https://ratchakitcha.soc.go.th/documents/17165183.pdf. (in Thai)

Yang XS. Nature-inspired optimization algorithms. Amsterdam: Morgan Kaufmann; 2014.

Yang XS, Deb S. Cuckoo search via Lévy flights. Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC); Coimbatore; India. 2009. p. 210-214.

Zaini N, Ean LW, Ahmed AN, Malek MA, Chow MF. PM2.5 forecasting for an urban area based on deep learning and decomposition method. Scientific Reports. 2022; 12: 17565, 1-13.

Zhao L, Li Z, Qu L. Forecasting of Beijing PM2.5 with a hybrid ARIMA model based on integrated AIC and improved GS fixed-order methods and seasonal decomposition. Heliyon. 2022; 8(12): e12239, 1-16.