Scientific Reports

https://doi.org/10.1038/s41598-026-48175-1

Article in Press

Enhancing Winters exponential smoothing: a novel approach with back-casting, Grey Wolf Optimization, and error tracking

Received: 24 September 2025

Accepted: 6 April 2026

Published online: 24 April 2026

Cite this article as: Karakullukcu E. Enhancing Winters exponential smoothing: a novel approach with back-casting, Grey Wolf Optimization, and error tracking. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-48175-1

Erdinc Karakullukcu

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

ARTICLE IN PRESS

© The Author(s) 2026. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

ARTICLE IN PRESS

Enhancing Winters exponential smoothing: A novel approach with back-casting, grey wolf optimization, and error tracking

Erdinc Karakullukcu[1*]

[1*]Department of Computer Sciences, Karadeniz Technical University, Universite Mahallesi, Trabzon, 61080, Trabzon, Turkey.

Corresponding author(s). E-mail(s): erdinc.karakullukcu@ktu.edu.tr;

Abstract

Winters Exponential Smoothing (WES) is a widely used method for forecasting time series data, particularly for handling seasonality in both additive and multiplicative models. This study aims to enhance the forecasting performance of the WES method by introducing three key modifications: back-casting for accurate initialization of WES components, the Grey Wolf Optimization (GWO) algorithm for automated parameter tuning, and a sequential error tracking signal (SETS) for adaptive bias correction. Beyond preliminary validations on German temperature data and the Time Series Data Library (TSDL), this research presents an extensive robustness analysis using 40 diverse series from the M4 Forecasting Competition across multiple temporal frequencies (quarterly, monthly, weekly, and hourly). To ensure methodological rigor and mitigate sampling bias, a rolling-origin evaluation protocol and a systematic GWO stability analysis are employed. Furthermore, the model's real-time adaptability is demonstrated through a one-step-ahead forecasting framework. Experimental results and ablation studies show that the proposed method achieves a superior global mean rank (1.475–1.625), significantly outperforming Auto-ARIMA, ETS, and Seasonal Naive models. Computational complexity and GWO convergence analyses further confirm that the proposed framework is significantly faster than automated ARIMA models while providing robust, high-precision forecasts suitable for large-scale and high-frequency applications.

Keywords: Winters exponential smoothing, Back-casting, Grey wolf optimization, Forecasting

ARTICLE IN PRESS

ARTICLE IN PRESS

1 Introduction

Winters Exponential Smoothing (WES) (Winters 1960) is a powerful statistical method for forecasting time series data exhibiting both trend and seasonal patterns. Its versatility has led to widespread application across various domains, including environmental studies (e.g., CO_2 emission prediction (Alam and AlArjani 2021) and groundwater level prediction (Sarma and Singh 2022)), health (Cañedo et al. 2024), cloud resource provisioning (Shahin 2016), and weather forecasting (Kinast and Fokoué 2024).

A significant challenge in implementing the WES method lies in determining suitable initial values for the level, trend, seasonal indices, and smoothing parameters (Gardner Jr 2006). Several approaches have been proposed to address this issue (Gardner Jr and McKenzie 1985; Hansun 2017; Hyndman et al. 2002; Wongoutong 2021). The back-casting technique, employed by Makridakis et al. (1982) in a forecasting competition, is regarded as a sophisticated method and a key component of the WES algorithm (Segura and Vercher 2001), and it is widely implemented in statistical software packages like SAS and SPSS (Yaffee and McGee 2000). Predictive performance is frequently enhanced by combining meta-heuristics with domain-specific architectures. Such hybrid approaches have been successfully implemented in diverse areas, ranging from wind speed forecasting systems (El-Kenawy et al. 2023) to the prediction of metamaterial antenna parameters (El-Kenawy et al. 2022), where optimization ensures the stability of the parameter space.

The proposed algorithm seeks to improve the forecasting performance of the WES on both additive and multiplicative time series by employing this back-casting approach to estimate the initial level, trend, and seasonal components within the WES method.

The WES method can be further improved by optimizing the smoothing parameters to minimize the sum of squared, one-step-ahead prediction errors. This has been demonstrated by various studies. Jiang et al. (2020) proposed a model integrating the Fruit Fly Optimization Algorithm (FOA) to perform WES for accurate medium-term electricity consumption forecasting, particularly when there is insufficient historical data available. Similarly, in the study by Kristianto (2019), FOA was used to optimize the smoothing parameters of WES for forecasting datasets on lung disease sufferers, traffic accidents, cow milk production, and water use. The study by Mauricio and Ostia (2023) utilized Holt-Winter forecasting method and compared the Cuckoo search algorithm (CSA) with the Genetic algorithm in optimizing Mean Squared Error (MSE) to obtain the optimum smoothing coefficients of WES.

At this point, a key innovation of the proposed method is the integration of Grey Wolf Optimization (GWO) to optimize the smoothing parameters (α, β, and γ) for both back-casting and forecasting phases of WES.

GWO, developed by Mirjalili et al. (2014), is a nature-inspired metaheuristic algorithm that mimics the social hierarchy and hunting behavior of grey wolves to search for optimal solutions and is recognized as one of the fastest-growing swarm intelligence algorithms.The integration of nature-inspired algorithms is a dominant trend for overcoming the limitations of static models. While GWO is utilized in this study for its robust convergence properties, the efficacy of other recent developments, such

ARTICLE IN PRESS

as the Binary Waterwheel Plant Optimization (BWPO) (Alhussan et al. 2023) and the Dipper Throated Optimization (DTO) (Takieldeen et al. 2022), underscores the necessity of adaptive search mechanisms in modern computational tasks.

The success of GWO has motivated researchers to apply it across diverse fields, including engineering (Liu et al. 2024), machine learning (Medjahed et al. 2016; Mirjalili 2015; Eswaramoorthy et al. 2016), medical and bioinformatics (Li et al. 2017), networking (Al-Aboody and Al-Raweshidy 2016), environmental applications (Song et al. 2015), image processing (Bilal et al. 2024), and time series forecasting (Tak 2022; El-Kenawy et al. 2023).

To the best of our knowledge, GWO has not been employed for the optimization of smoothing parameters of WES so far. It is worthwhile to know how to implement GWO algorithm and to compare the effectiveness of the GWO with that of WES.

Given the dynamic nature of time series forecasting, models often need updates after certain time periods (Deng et al. 2004). Monitoring is vital for accurate forecasting, as it identifies deviations and the need for corrective actions to maintain control over the forecasting process. Tracking signal is a common statistic that utilizes forecast errors to monitor the system's performance. First proposed by Brown (1959) and later analyzed by Trigg (1964) and Gardner Jr (1985), tracking signals come in generally two forms: Cumulative Sum Tracking Signal (CTS) and Smoothed Error Tracking Signal (SETS). This research focuses on SETS due to its high sensitivity to recent forecast errors, proving more effective in previous applications (Nembhard and Kao 2001). Yu et al. (2005) proposed an adaptive smoothing neural network to predict foreign exchange rates, where adaptive smoothing techniques were used to adjust the neural network learning parameters automatically by tracking signals under dynamic varying environments. Moreover, Bianchesi et al. (2022) presented a nonlinear time series prediction methodology using neural networks and tracking signals method to detect bias and their responsiveness to non-random changes in the time series. To our knowledge, although tracking signals are common in neural network-based forecasting, no existing work addresses their application with WES.

To enhance the robustness and accuracy of the model, SETS is incorporated into the proposed algorithm. This mechanism monitors forecast errors and triggers adjustments when necessary, such as when the model consistently overestimates or underestimates actual values. By continuously adapting to changing data patterns, the model maintains its predictive accuracy over time.

This paper presents a novel algorithm that leverages the WES method, GWO, and SETS to enhance the forecasting accuracy. Back-casting is used to initialize the WES model, and GWO optimizes its parameters (α, β, and γ). SETS is incorporated to ensure the model's ability to adapt to evolving data patterns.

The paper is structured as follows: Sect. 2 introduces the WES, GWO, and SETS methods. Sect. 3 presents and discusses the proposed approach. Sect. 4 describes the experimental results and the performance of the proposed method. Sect. 5 addresses critical methodological considerations, such as potential sampling bias in fixed data splits, the generalizability of the model across diverse frequencies, and the stability of the metaheuristic optimization process. Finally, conclusions are drawn in Sect. 6.

ARTICLE IN PRESS

2 Methods

2.1 Winters exponential smoothing

The Holt-Winters family of exponential smoothing methods, developed in the 1960s, has played a central role in time series forecasting and signal processing (Trull et al. 2020). Winters (1960) introduced what is now known as the multiplicative Holt-Winters model (or triple exponential smoothing), which remains one of the most widely adopted approaches for modeling seasonal data. This method is particularly suitable when seasonal fluctuations increase or decrease proportionally with the overall level of the series. Alternatively, the additive Holt-Winters model is applied when the seasonal effect remains approximately constant across time.

The multiplicative version consists of three recursive smoothing equations, corresponding to the level, trend, and seasonal components, along with a forecasting function, as expressed in Equations (1)–(4) (IBM SPSS 2016):

$$L_t = \alpha \left(\frac{Z_t}{S_{t-s}} \right) + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{1}$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \tag{2}$$

$$S_t = \gamma \left(\frac{Z_t}{L_t} \right) + (1-\gamma)S_{t-s} \tag{3}$$

$$\hat{Z}_t(k) = (L_t + kT_t)S_{t+k-s} \tag{4}$$

where α, β, and γ represent the smoothing parameters, while L_t, T_t, and S_t denote the level, trend, and seasonal components, respectively. The observed values (Z_t) are iteratively decomposed into these three components. Forecasts are then obtained by extrapolating the smoothed components k steps ahead using Equation (4), yielding predictions ($\hat{Z}_t(k)$) (Trull et al. 2020).

Similar to the multiplicative model, the additive Winters model also consists of three recursive smoothing equations, corresponding to the level, trend, and seasonal components, along with a forecasting function, as expressed in Equations (5)–(8) (IBM SPSS 2016):

$$L_t = \alpha \left[Z_t - S_{t-s} \right] + (1-\alpha) \left(L_{t-1} + T_{t-1} \right) \tag{5}$$

$$T_t = \beta \left(L_t - L_{t-1} \right) + (1-\beta) \, T_{t-1} \tag{6}$$

$$S_t = \gamma \left(Z_t - L_t \right) + (1-\gamma) \, S_{t-s} \tag{7}$$

$$\hat{Z}_t(k) = L_t + kT_t + S_{t+k-s} \tag{8}$$

In the additive Winters model, the seasonal factors S_{t-s} are normalized so that they sum to zero, whereas in the multiplicative Winters model, the seasonal factors

ARTICLE IN PRESS

are normalized to have an average of one (Yaffee and McGee 2000). For both the additive and multiplicative Holt-Winters formulations, the smoothing parameters are typically estimated by minimizing the sum of squared one-step-ahead forecast errors (IBM SPSS 2016)

2.1.1 Initialization of exponential smoothing

In the Holt-Winters framework, L denotes the level state, T the trend state, and S, a vector of length s, the seasonal states. For the additive Winters model, in the back-casting step, the initial value for the level is the last observation of the original time series. To obtain the initial value for the trend component, $Z = \alpha t + \sum_{i=1}^{s-1} \beta_i I_i(t)$ is fit to the data where t is time and $I_i(t)$ are seasonal dummies (Note that the model includes intercept). Then, $T = -\alpha$, and $S = \beta - mean(\beta)$ (Note that, since the number of β parameters is $s-1$, the last seasonal parameter is $-mean(\beta)$, so that the factors sum to zero). It is important that all smoothing calculations in the back-casting step are performed on the time-reversed data (from $t = n$ to $t = 0$) (IBM SPSS 2016).

Following the back-casting process, adjustments are necessary to initialize the parameters T, L, and S for forecasting step. To ensure the seasonal factors obtained at the end of the back-casting stage (S_{t-s}) sum to 0, the mean of the seasonal factors is subtracted from each seasonal factor. The initial value for the level is calculated by summing the mean of the seasonal factors with the trend and level values from the final back-casting step. The initial trend value is calculated by taking the negative of the trend value from the final back-casting step. Then, the initial smoothing states for forecasting are as follows (IBM SPSS 2016).

$$S' = S_{t-s} - \text{mean}(S_{t-s}) \tag{9}$$

$$T' = -T_n \tag{10}$$

$$L' = L_n + T_n + \text{mean}(S_{t-s}) \tag{11}$$

Algorithm 1 provides the pseudocode for the additive WES method.

ARTICLE IN PRESS

ARTICLE IN PRESS

Algorithm 1 Additive WES Pseudocode

Require: Time series dataset (Z), length of seasonality (s), number of periods in the forecast lead period (k)

Ensure: Forecasts for k periods into the future

Back-casting Step

1: Set the initial value for the level as the last observation of the original time series
$L = Z_n$
2: $Z = \alpha t + \sum_{i=1}^{s} \beta_i l_i(t)$ is fit to the data where t is time and $l_i(t)$ are seasonal dummies.
$T = -\alpha$
3: The initial value for the seasonal component
$S = \beta - mean(\beta)$
4: Perform smoothing calculations on the time-reversed data (from $t = n$ to $t = 0$)
5: Employ GWO to optimize the smoothing parameters α, β, and γ, aiming to minimize RMSE
6: Update the level component
$L_t = \alpha[Z_t - S_{t-s}] + (1 - \alpha)(L_{t-1} + T_{t-1})$
7: Update the trend component
$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$
8: Update the seasonal component
$S_t = \gamma(Z_t - L_t) + (1 - \gamma)S_{t-s}$
9: Obtain the predictions
$\hat{Z}_t(k) = L_t + kT_t + S_{t+k-s}$

Forecasting Step

10: Make adjustments to set the initial values for the forecasting step
11: Ensure the seasonal factors obtained at the end of the back-casting stage (S_{t-s}) sum to 0
$S' = S_{t-s} - mean(S_{t-s})$
12: $L' = L_n + T_n + mean(S_{t-s})$
13: $T' = -T_n$
14: Apply Steps 6, 7, 8, and 9 to generate forecasts using the obtained initial values
15: **Return** the forecasts

ARTICLE IN PRESS

For the multiplicative Winters model, the initial value for the level is the last observation of the original time series. The initial trend component is estimated by fitting a separate regression line (including an intercept) for each season, using time as the predictor (IBM SPSS 2016). Let μ denote the vector of intercepts and β the vector of slopes, both of length s. Then,

$$T = -mean\,(\beta) \tag{12}$$

$$L = Z_n \tag{13}$$

$$S = \frac{\mu + \beta}{mean\,(\mu) + mean\,(\beta)} \tag{14}$$

ARTICLE IN PRESS

It is important to note that all smoothing calculations in this analysis are performed on the time-reversed data (from $t = n$ to $t = 0$). Following the back-casting process, adjustments are necessary to initialize the parameters T, L, and S for forecasting step. To ensure the seasonal factors obtained at the end of the back-casting stage (S_{t-s}) average to 1, each seasonal factor is divided by the mean of the seasonal factors. The initial value for the level is calculated by multiplying the mean of the seasonal factors by the sum of the trend and level values from the final back-casting step. The initial trend value is calculated by multiplying the negative of the final back-casting trend value by the mean of the seasonal factors. Then, the initial smoothing states for forecasting are as follows (IBM SPSS 2016).

$$S' = \frac{S_{t-s}}{\text{mean}(S_{t-s})} \tag{15}$$

$$T' = -T_n \times \text{mean}(S_{t-s}) \tag{16}$$

$$L' = (L_n + T_n) \times \text{mean}(S_{t-s}) \tag{17}$$

Algorithm 2 provides the pseudocode for the multiplicative WES method.

ARTICLE IN PRESS

ARTICLE IN PRESS

Algorithm 2 Multiplicative WES Pseudocode

Require: Time series dataset (Z), length of seasonality (s), number of periods in the forecast lead period (k)

Ensure: Forecasts for k periods into the future

Back-casting Step

1: Set the initial value for the level as the last observation of the original time series
$L = Z_n$
2: Fit a separate line for each season with time as a regressor. Suppose μ is the vector of intercepts and β is the vector of slopes, then the initial value for the trend component
$T = -\text{mean}(\beta)$
3: The initial value for the seasonal component
$S = \frac{\mu+\beta}{\text{mean}(\mu)+\text{mean}(\beta)}$
4: Perform smoothing calculations on the time-reversed data (from $t = n$ to $t = 0$)
5: Employ GWO to optimize the smoothing parameters α, β, and γ, aiming to minimize RMSE
6: Update the level component
$L_t = \alpha \left[\frac{Z_t}{S_{t-s}}\right] + (1-\alpha)(L_{t-1} + T_{t-1})$
7: Update the trend component
$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$
8: Update the seasonal component
$S_t = \gamma\left(\frac{Z_t}{L_t}\right) + (1-\gamma)S_{t-s}$
9: Obtain the predictions
$\hat{Z}_t(k) = (L_t + kT_t)S_{t+k-s}$

Forecasting Step

10: Make adjustments to set the initial values for the forecasting step
11: Ensure the seasonal factors obtained at the end of the back-casting stage (S_{t-s}) average to 1
$S' = \frac{S_{t-s}}{\text{mean}(S_{t-s})}$
12: $L' = (L_n + T_n) \times \text{mean}(S_{t-s})$
13: $T' = -T_n \times \text{mean}(S_{t-s})$
14: Apply Steps 6, 7, 8, and 9 to generate forecasts using the obtained initial values
15: **Return** the forecasts

ARTICLE IN PRESS

2.1.2 WES implementation in SPSS, R, and Python

When examining the functions related to the WES method for seasonal series in statistical software packages, it is observed that the formulas for level, trend, and seasonality used for both additive and multiplicative series are the same. However, there are differences between the forecasts obtained using these methods. The reasons for these differences include the initial values assigned to the smoothing parameters, the type of objective function to be minimized (such as MSE, Mean Absolute Error (MAE), etc.), the optimization method used to determine the smoothing parameters

ARTICLE IN PRESS

that yield the lowest objective function value, and the selection of initial values for level, trend, and seasonality.

Table 1 provides information on the default values used in the implementation of the WES method in SPSS, R, and Python, as well as how these values are applied. Unlike the initial values and optimization methods used in the statistical packages listed in the table, the forecasting results for the datasets presented in the Experimental Results section were obtained by implementing the GWO method in the optimization phase of the WES method, incorporating the back-casting method to determine the initial values of level, trend, and seasonality, and further applying the proposed monitoring phase.

2.2 Grey wolf optimizer algorithm

The Grey Wolf Optimizer (GWO) is a nature-inspired optimization technique that mimics the hierarchical social organization and cooperative hunting strategies of grey wolves in the wild. Introduced by Mirjalili et al. (2014), GWO is a modern addition to swarm intelligence algorithms. Its simplicity, adaptability, and efficiency in tackling complex optimization challenges have garnered it considerable attention and use across various fields.

GWO takes cues from the hierarchical organization and cooperative hunting tactics of grey wolves, renowned for their effective pack dynamics and prey-capturing prowess. In the wild, grey wolves follow a strict social hierarchy consisting of four main groups: alpha, beta, delta, and omega. Alpha (α) wolves lead the pack, making vital decisions, while beta (β) wolves support and advise the alphas. Delta (δ) wolves are responsible for the pack's protection, and omega (ω) wolves, the lowest-ranking members, help diffuse tension within the pack by absorbing aggression. This hierarchical structure, which is crucial to the operation of the GWO algorithm, is illustrated in Figure 1. In the design of the GWO algorithm, the best solution is designated as alpha (α) to mathematically model the wolves' social hierarchy. The second and third best solutions are designated as beta (β) and delta (δ), respectively, while the remaining candidate solutions are categorized as omega (ω). The search process is primarily directed by the α, β, and δ wolves, with the ω wolves adapting their positions in response to these leading groups (Mirjalili et al. 2014).

Beyond social hierarchy, GWO algorithm emulates the collective hunting strategy of grey wolf packs, which is structured in three phases: searching for, pursuing, and finally attacking prey. This biological process is mapped to the optimization framework as exploration (global search), intensification (local search), and convergence (final attack), enabling the algorithm to progressively refine solutions and approach the global optimum (Mirjalili et al. 2014). The pseudocode in Algorithm 3 outlines the basic steps of the GWO algorithm. The process of the GWO algorithm is as follows (Mirjalili et al. 2014):

Step 1 Set the population size (N), maximum iterations (T_{max}), individual grey wolf dimension (dim), and parameters α, A, and C. Initialize each member using Equation (18).

ARTICLE IN PRESS

Table 1 The default values used in the implementation of the WES method in SPSS, R, and Python

R (stats package, HoltWinters function)	
Component	**Value**
Initial Alpha	0.3
Initial Beta	0.1
Initial Gamma	0.1
Optimization Algorithm	Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Box constraints (L-BFGS-B) (via optim())
Initial Level and Initial Trend	Time series data is decomposed into trend, seasonal, remainder, and figure (a combination of the seasonal and trend components). The portion of the time series used in the decomposition step is the first x data points, where x is the number of periods (default is 2), multiplied by the frequency of the time series. The trend component is extracted from the decomposition result, and any missing values are removed. Linear regression is then performed on the resulting data, where a constant term and a sequence of integers representing time are added. This regression allows for the estimation of initial level and trend. The linear regression is carried out, and the coefficients are extracted. The first coefficient is interpreted as the intercept, corresponding to the initial level, while the second coefficient is interpreted as the slope, corresponding to the initial trend (Rdocumentation 2024; Winters 1960).
Initial Seasonality	The combination of the trend and seasonal components is used for the initial values of the seasonal component.
PYTHON (ExponentialSmoothing class in statsmodels.tsa.holtwinters module)	
Component	**Value**
Initial Alpha	$0.5/\max(m, 1)$, where m is the seasonal period
Initial Beta	$0.1 \times alpha$
Initial Gamma	$0.05 \times (1 - alpha)$
Optimization Algorithm	L-BFGS-B (via scipy.optimize.minimize())
Initial Level and Initial Trend	Time series data is first indexed using an array of integers, which represent the time points of the observations. Then, a matrix is constructed, where the first column contains the time indices and the second column consists of ones, representing the intercept in the linear regression. A least squares linear regression is performed, with the time indices as the independent variable and the observed values as the dependent variable. This regression process estimates the initial level (intercept) and initial trend (slope) by solving for the coefficients that best fit the time series data, where the level represents the starting value of the series, and the trend reflects the rate of change in the series (Seabold and Perktold 2010).
Initial Seasonality	The data is reshaped into a matrix where each row corresponds to one full seasonal cycle. Then, the mean for each season (e.g., January, February, etc.) is computed. Next, these seasonal means are normalized so that their overall mean equals 1. The normalized seasonal components are then used as the initial seasonal values (Seabold and Perktold 2010).
SPSS (Winters Exponential Smoothing within the Time Series Modeler)	
Component	**Value**
Initial Alpha	Not explicitly documented.
Initial Beta	Not explicitly documented.
Initial Gamma	Not explicitly documented.
Optimization Algorithm	Not explicitly documented.
Initial Level and Initial Trend	Given in detail in Sect. 2.1.1.
Initial Seasonality	Given in detail in Sect. 2.1.1.

ARTICLE IN PRESS

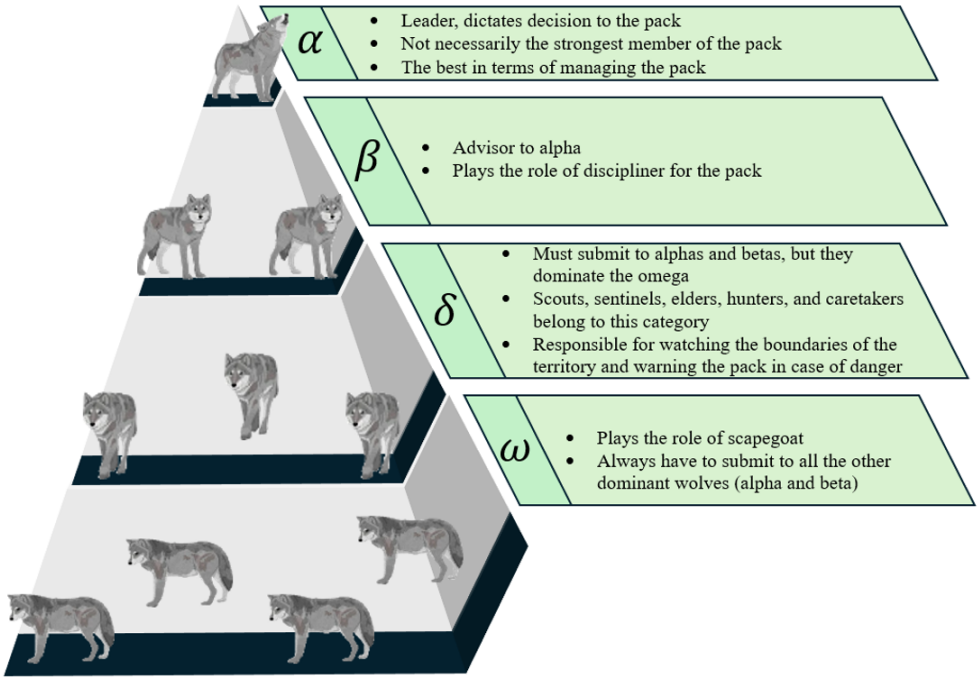


Fig. 1 Hierarchy of the grey wolf pack (dominance decreases from top to bottom)

$$X = rand\,(0, 1, N, dim) * (UL - LB) + LB \tag{18}$$

where LB and UB denote the lower and upper bounds of the solution space, respectively. X represents the positions of the current solution. $rand$ is a stochastic function that creates a N by dim array (or matrix) filled with random values uniformly distributed between 0 and 1.

Step 2 Evaluate the fitness value of each individual using the test function. Then, based on the fitness values, designate the best-fitting individual as the α-wolf, the second-best as the β-wolf, and the third-best as the δ-wolf.

Step 3 The mathematical model for the grey wolf hunting process is described by Equation (19). This equation calculates the overall direction of movement for the pack based on the distance difference between the leader and the other wolves. The movement direction information for each individual wolf can be calculated using Equations (20)–(21). To update the current position of each grey wolf, Equations (19)–(21) are applied.

ARTICLE IN PRESS

Algorithm 3 Pseudocode of the GWO

Initialize the population of grey wolves X_i $(i = 1, ..., n)$
Initialize a, A, and C
Calculate the fitness of each search agent in the population
X_α is the best search agent, X_β is the second-best search agent, X_δ is the third-best search agent
Set iteration counter (t) to 1
Set the maximum number of iterations T_{max}
while $t < T_{max}$ **do**
    for each search agent **do**
        Update the score of leader wolves α, β, and δ
        Update the position of the leader wolves X_α, X_β, and X_δ
        Evaluate new position's fitness
    end for
    Update a using Equation (18)
    for each search agent **do**
        Update A and C using Equation (19)
        Calculate the distance between current individuals X_i and leader wolves X_α, X_β, X_δ using Equation (15)
        Update X_1, X_2, X_3 using Equation (16)
        Update the position of the current search individual X_{i+1} using Equation (4)
    end for
    Calculate the fitness of all search agents in the population
    $t = t + 1$
end while
Return the best solution X_α

$$\begin{aligned} D_\alpha &= |C_1 X_\alpha - X| \\ D_\beta &= |C_2 X_\beta - X| \\ D_\delta &= |C_3 X_\delta - X| \end{aligned} \tag{19}$$

$$\begin{aligned} X_1 &= X_\alpha - A_1 D_\alpha \\ X_2 &= X_\beta - A_2 D_\beta \\ X_3 &= X_\delta - A_3 D_\delta \end{aligned} \tag{20}$$

$$X\,(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{21}$$

The symbols D_α, D_β, and D_δ denote the distance vectors between the alpha, beta, and delta wolves and the remaining wolves in the population. Similarly, X_α, X_β, and X_δ correspond to the positions of the alpha, beta, and delta wolves. The random coefficient vectors C_1, C_2, C_3 and A_1, A_2, A_3 are generated in accordance with the constraints defined in Equation (23). Based on these quantities, X_1, X_2, and X_3 represent the position updates influenced by the three leading wolves, while the overall

ARTICLE IN PRESS

movement of the pack at the next iteration is expressed as $X(t+1)$. As illustrated in Figure 2, the updated positions of wolves are distributed within a region bounded by the locations of the alpha, beta, and delta wolves. This illustrates how the leadership hierarchy governs the spatial dynamics of the population and directs the collective pursuit of prey (Wang and Li 2019)

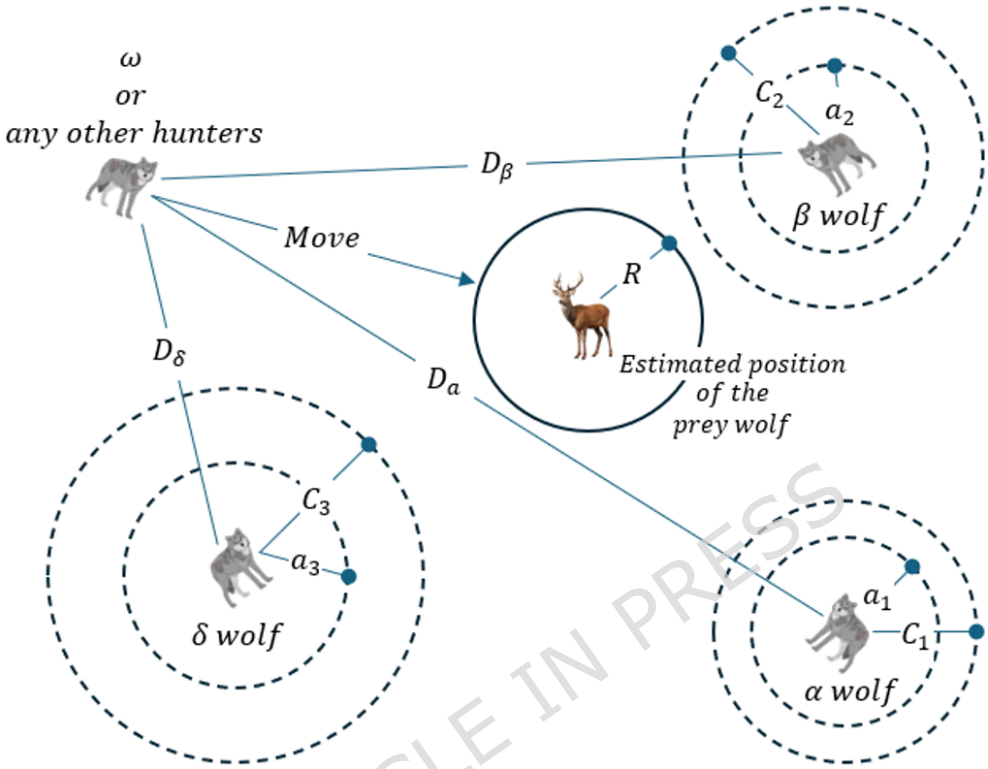


Fig. 2 Position updating mechanism of the wolves in 2D space

Step 4 Update a, A, and C using Equations (22)–(23).

$$a = 2\left(1 - \frac{t}{T_{max}}\right) \tag{22}$$

$$\begin{cases} A = 2ar_1 - a \\ C = 2r_2 \end{cases} \tag{23}$$

where t denotes the current iteration, T_{max} represents the maximum number of iterations, and r_1 and r_2 are random vectors uniformly sampled from the interval [0,1]. The parameter 'a' in this algorithm decreases linearly from 2 to 0 over successive iterations. This models the behavior of the wolves as they approach their prey (exploration

ARTICLE IN PRESS

phase). A higher initial value of a promotes rapid global exploration, directing the wolf pack towards promising regions of the search space. As the algorithm progresses, reducing a encourages a more focused search near the optimal solution, leading to improved convergence accuracy. Figure 3 illustrates a 2D position vector along with some possible neighboring positions. As illustrated, a grey wolf located at position (X, Y) updates its position with respect to the prey?s location (X^*, Y^*). By varying the values of the A and C coefficient vectors, the wolf can explore alternative locations in the vicinity of the best solution. This mechanism naturally generalizes to an n-dimensional search space, where wolves navigate within hypercubes or hyperspheres centered around the current best position. The inequality $|a| < 1$ forces wolves to attack the prey, whereas the inequality $|a| > 1$ forces the wolves to scatter in the environment to find a better prey (Mirjalili et al. 2014).

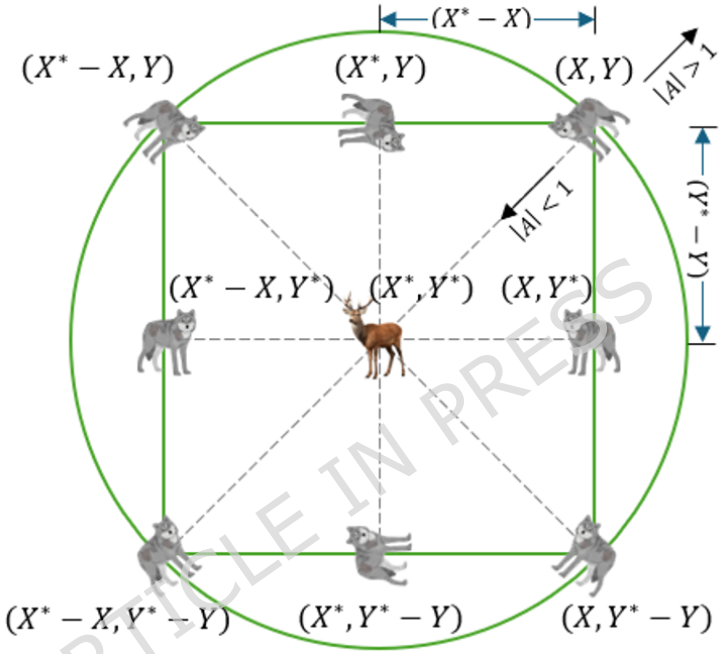


Fig. 3 2D position vectors and their possible next locations

Step 5 Adjust the positions of the non-leader wolves. Assess the fitness of the new positions. Update the a, β, δ-wolves, and the global optimal solution (R in Figure 2 denotes the position vector of the optimization target).

Step 6 Determine if the stopping criterion has been met (e.g., maximum iterations reached). If not, repeat Steps 2 to 5. Once the stopping criterion is met, the final position of the alpha wolf represents the optimal solution, and its corresponding fitness value indicates the solution's quality.

ARTICLE IN PRESS

2.3 Monitoring

Monitoring is a critical component of time series forecasting, especially when using intelligent methods. A robust monitoring system involves real-time tracking of forecast errors and diagnosing potential issues with model parameters. When non-random changes are detected, the model must be promptly adjusted. Given the dynamic nature of time series forecasting, it is common for the model to require updates after a certain period. However, the questions of when and how to update the model parameters remain unresolved. Despite its significance, this aspect of intelligent forecasting has received limited attention in research (Deng et al. 2004).

2.3.1 Tracking Signals

The primary goal of forecasting is to generate accurate and unbiased predictions. A reliable forecasting model is essential to achieve this goal. Forecast error, calculated as the difference between actual and predicted values (Equation (24)), is a key indicator of model performance. A major disparity between a forecast and the actual result can substantially raise a company?s operating costs while also creating missed opportunities (McClain 1988).

$$e_t = Z_t - \hat{Z}_t \tag{24}$$

where e_t is the forecast error at time t; Z_t and $\hat{Z}_t$ are the actual and forecasted values at time t, respectively.

Cumulative Forecast Error (CFE) and Mean Absolute Deviation (MAD) are two commonly used metrics for assessing forecast accuracy. These metrics are often employed to evaluate the performance of forecasting models (Wisner et al. 2009).

$$CFE_t = \sum_{i=1}^{t} e_i \tag{25}$$

$$MAD_t = \frac{\sum_{i=1}^{t} |e_i|}{t} \tag{26}$$

While CFE measures the overall bias of forecasts, the tracking signal is a valuable tool for detecting and monitoring forecast bias. CFE indicates whether forecasts are consistently overestimated or underestimated, while tracking signal helps identify when a significant deviation from the expected pattern occurs (Wisner et al. 2009).

The Smoothed Error Tracking Signal (SETS) (Trigg 1964) is one of the most frequently used tracking signals proposed for monitoring forecasting model performance (Li et al. 2012).The formulation of SETS is given in Equation (27).

$$T_t = \left| \frac{E_t}{SMAD_t} \right| \tag{27}$$

$$E_t = \alpha_e e_t + (1 - \alpha_e) E_{t-1}, \qquad 0 \leq \alpha_e \leq 1 \tag{28}$$

ARTICLE IN PRESS

$$SMAD_t = \alpha_m |e_t| + (1 - \alpha_m) SMAD_{t-1}, \qquad 0 \leq \alpha_m \leq 1 \tag{29}$$

where T_t is the smoothed error tracking signal, E_t is the smoothed value of forecast errors, $SMAD_t$ is the smoothed MAD at period t, α_e is the error smoothing parameter ($0 \leq \alpha_e \leq 1$), e_t is the forecast error at time t, α_m is the MAD smoothing parameter ($0 \leq \alpha_m \leq 1$).

The initial value for E_t equals zero and the initial value for MAD_t is set to the mean absolute deviation of training data (Deng et al. 2004).

A control limit must be predetermined to evaluate the tracking signal (T_t). The choice of control limit depends on the smoothing parameters (α_e and α_m). It is important to note that the optimal smoothing parameters vary based on the specific data being analyzed. For sudden changes in the system, larger smoothing parameters are needed to prioritize recent observations. Conversely, for relatively stable systems, smaller smoothing parameters can be used to give more weight to historical data. Traditionally, the SETS model has applied equal values for the smoothing parameters in both the numerator and denominator, meaning α_e=α_m. However, McClain (1988) proposed that the smoothing parameter for the MAD, α_m, should be set lower than that of the numerator, α_e, in order to help stabilize the variance of observations. α_e and α_m are typically selected between 0.05 and 0.15 (Nembhard and Chen 2007).

To determine if a process is out of control, a tracking signal is compared to a predetermined threshold (K_S). A common range for K_S is 0.4 to 0.6 (Montgomery et al. 1990). If the tracking signal exceeds K_S, it indicates a potential issue with the forecasting model, suggesting that the one-step-ahead prediction errors may be biased.

3 Proposed method

The proposed algorithm aims to enhance forecast accuracy for time series suitable for multiplicative or additive WES. To achieve this, the algorithm employs the back-casting method to obtain initial estimates for the level, trend, and seasonal components within the WES method. By iteratively working backward from the most recent data point, this method provides more accurate initial estimates for level, trend, and seasonal components compared to arbitrary initializations. Thus, improved forecasting accuracy can be achieved particularly for time series with pronounced trends or seasonal patterns. Additionally, back-casting can accelerate convergence, reduce computational costs, and result in more unbiased forecasts.

The initial estimates obtained are then used in the forecasting step of the Winters method. A key modification to improve the algorithm's performance in both back-casting and forecasting steps is the utilization of GWO. This method enhances the selection of α, β, and γ parameters in Winters exponential smoothing by using an optimization technique inspired by the social structure and hunting behaviors of grey wolves. Through iterative adjustments, GWO seeks parameter values that minimize the Root Mean Squared Error (RMSE). GWO's global exploration capabilities also reduce the risk of becoming trapped in local minima and enable efficient convergence to optimal parameter values.

ARTICLE IN PRESS

The final step of the algorithm involves the use of SETS, which enhances the model's capability to detect and adapt to changes in the underlying data patterns. SETS aids in identifying instances where forecasts consistently overestimate or underestimate actual values. When the tracking signal surpasses a defined threshold, it indicates that the model's parameters may need adjustment to improve forecast accuracy. By monitoring forecast errors and triggering adjustments as needed, the model can maintain accuracy and reliability over time.

The proposed algorithm (Algorithm 4) uses time series data ($original_data$) as input. The additive or multiplicative WES method is employed for forecasting. The data is split into training ($train$), validation (val), and testing sets ($test$). The first 60% of the time series is used for training, the subsequent 30% for validation, and the remaining 10% for testing.

The period ($period$) is set in accordance with the seasonality of the time series. The recommended values for the error smoothing parameter (α_e) and the MAD smoothing parameter (α_m) are between 0.05 and 0.15 (Nembhard and Chen 2007). Since better results are expected when α_m is smaller than α_e, iterative processes are performed for various combinations of these parameters from the set $\{0.05,\ 0.10,\ 0.15\}$.

Using the appropriate WES method (Algorithms 1 or 2), forecasts are obtained for the training and validation sets ($train_predictions$ and $val_forecasts$, respectively). To determine the optimal smoothing parameters for WES, GWO (Algorithm 3) is employed. In GWO, the lower and upper bounds for the α, β, and γ parameters are set to 0 and 1, respectively ($lb = [0,\ 0,\ 0]$, $ub = [1,\ 1,\ 1]$). The individual grey wolf dimension (dim) is 3, the number of search agents ($n_searchAgents$) is 10, and the maximum number of iterations (T_{max}) is 50 (These hyperparameters were selected based on their perceived reasonableness, but they can be further optimized through hyperparameter tuning). An array is created to store the monitored validation forecasts ($mon_val_forecasts$).

Errors for the training set ($train_error$) and validation set (val_error) are calculated. Equations (9)–(10) are used to obtain the smoothed value of validation errors ($smoothed_errors$) and the smoothed MAD values ($SMAD$). The initial value for MAD_t is set to the mean absolute deviation of the training data. Using Equation (27), the smoothed error tracking signal ($error_tracking_signal$) is obtained. The first index at which three consecutive absolute values of this signal exceed the chosen threshold value of 0.6 is considered the point where bias in the validation set deteriorates. This index is stored in a variable ($index$).

The portion of the validation set up to this index is added to the training set, and an equal-sized portion from the beginning of the current training set is removed. The part added to the training set is removed from the validation set, updating the validation set. This process of adding and removing data from the training set continues until the smoothed error tracking signal is no longer large enough to cause bias. During this process, the forecasts obtained for the validation set are accumulated in $mon_val_forecasts$.

From the last point reached after monitoring, the series up to the end point of the validation data is subjected to the WES process according to the obtained optimal α, β, and γ parameters, and forecasts are generated for the test set ($test_forecasts$).

ARTICLE IN PRESS

Algorithm 4 Pseudocode for the proposed method

Require: Time series data (*original_data*)
Ensure: Forecasted values (*test_forecasts*) and RMSE (*test_rmse*)

Use *original_data* as input
$n \leftarrow length(original_data)$
Split data: train (60%), val (30%), test (10%)
$train_end \leftarrow n \times 0.6$; $val_end \leftarrow n \times 0.9$
$train \leftarrow original_data[0 : train_end]$; $original_train \leftarrow train$
$val \leftarrow original_data[train_end : val_end]$; $original_val \leftarrow val$
$train_and_val \leftarrow original_data[val_end : n]$
Specify period (*period*)
for α_e in $\{0.05, 0.10, 0.15\}$ **do**
    for α_m in $\{0.05, 0.10, 0.15\}$ **do**
        if $\alpha_m \leq \alpha_e$ **then**
            Use WES model (Algorithms 1, 2)
            Get *train_predictions*, *val_forecasts*
            Use GWO (Algorithm 3) to optimize the parameters of WES
            Init GWO population with Equation (18)
            Bounds for alpha, beta, gamma: $lb \leftarrow [0, 0, 0], ub \leftarrow [1, 1, 1]$
            $dim \leftarrow 3, n_searchAgents \leftarrow 10, T_{max} \leftarrow 50$
            Create the monitored validation forecasts array (*mon_val_forecasts*)
            $index \leftarrow 1, tracking_index \leftarrow 0$
            while $index > 0$ **do**
                $train_errors \leftarrow train - train_predictions$
                $val_errors \leftarrow val - val_forecasts$
                Obtain *smoothed_errors* using Equation (28)
                Obtain $SMAD$ using Equation (29)
                Obtain *error_tracking_signal* using Equation (27)
                Find *index* where 3 consecutive $error_tracking_signal > 0.6$
                if $index < 0$ **then**
                    Append *val_forecasts* to *mon_val_forecasts*
                    break
                end if
                Append *val_forecasts*[0 to index] to *mon_val_forecasts*
                $tracking_index \leftarrow tracking_index + index$
                Update datasets:
                $train \leftarrow train_and_val[tracking_index$ to $(train_size + tracking_index)]$
                $val \leftarrow train_and_val[(train_size + tracking_index)$ to $end]$
                Reapply WES+GWO to obtain *train_predictions* and *val_forecasts*
            end while
            $train \leftarrow train_and_val[tracking_index$ to $end]$
            Get *test_forecasts* using optimized WES
            Compute metrics using Equations (15)–(18)
            $val_rmse \leftarrow RMSE(original_val, mon_val_forecasts)$
            $test_rmse \leftarrow RMSE(original_test, test_forecasts)$
            Store results in *hyper_parameter_results*
        end if
    end for
end for
Using *hyper_parameter_results*, find α_e, α_m that minimize *val_rmse*
Return best *test_rmse* and *test_forecasts*

ARTICLE IN PRESS

ARTICLE IN PRESS

Using Equation (30), the RMSE values for the test and validation sets are obtained ($test_rmse$ and val_rmse, respectively). The current iteration's α_e, α_m, val_rmse, $test_rmse$, and $test_forecasts$ are stored in $hyper_parameter_results$. The same process is repeated for all possible combinations of α_e and α_m, and the parameter combination that minimizes val_rmse is determined. The forecasts ($test_forecasts$) obtained using this parameter combination, along with the corresponding $test_rmse$, constitute the output of the algorithm.

4 Evaluation

4.1 Experimental datasets

To assess the effectiveness of the proposed algorithm on additive time series, the mean values for the individual federal states and for Germany which are calculated based on gridded data for air temperature were adopted from the Climate Data Center (CDC) of the Deutscher Wetterdienst (DWD) (DWD 2024). The dataset covers the period from January 1881 to October 2024, encompassing all 16 federal states of Germany and Germany as a whole. It consists of monthly average temperature data. Table 2 presents the monthly minimum, maximum, and average temperatures for Germany and its states over the period 1881-2024. Figure 4 shows a map of Germany's federal states and the average temperatures observed in these states throughout the year 2023.

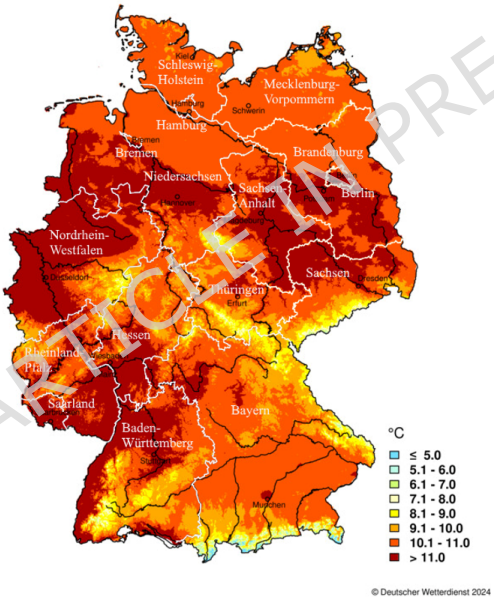


Fig. 4 Map of Germany showing the average air temperatures for Germany, January to December 2023 (map: German Weather Service, 2024; https://www.dwd.de/DE/leistungen/klimakartendeutschland/klimakartendeutschland.html

Table 2 Monthly minimum, maximum, and average temperatures for Germany and its states (1881-2024)

	Jan.	**Feb.**	**Mar.**	**Apr.**	**May**	**Jun.**	**Jul.**	**Aug.**	**Sep.**	**Oct.**	**Nov.**	**Dec.**
BB/BE	-10.2; 5.2	-11.5; 6.7	-2.2; 7.7	4.6; 12.9	9.8; 17.7	11.8; 21.9	14.8; 23.2	14.9; 21.3	9.9; 17.9	5.0; 12.6	-0.7; 7.8	-6.1; 6.8
	-0.3	0.4	3.6	8.3	13.9	16.7	18.4	17.7	14.0	9.1	4.2	1.1
BB	-1.2; 5.2	-11.5; 6.6	-2.2; 7.7	4.6; 12.9	9.7; 17.7	11.8; 21.9	14.8; 23.2	14.9; 21.3	9.9; 17.9	4.9; 12.6	-0.7; 7.8	-6.1; 6.8
	-0.4	0.4	3.6	8.3	14.4	16.7	18.4	17.7	14.0	9.1	4.2	1.1
BW	-7.8; 4.3	-10.2; 6.5	-0.8; 7.7	4.2; 12.5	8.3; 15.6	11.3; 20.9	13.8; 21.8	13.4; 21.7	8.3; 17.3	3.7; 12.9	-0.4; 7.6	-5.5; 5.1
	-0.6	0.5	3.9	7.8	12.3	15.6	17.3	16.7	13.3	8.5	3.7	0.5
BY	-9.7; 3.5	-11.3; 6.1	-2.4; 7.1	3.7; 12.5	8.0; 15.7	11.2; 20.3	13.6; 21.2	13.3; 21.2	7.8; 16.6	3.1; 11.9	-0.8; 6.8	-5.9; 4.3
	-1.7	-0.6	3.1	7.3	12.1	15.3	17.0	16.3	12.8	7.9	2.9	-0.4
HE	-9.6; 4.7	-9.4; 6.7	-1.5; 7.5	4.3; 12.6	8.4; 15.9	10.9; 19.5	13.9; 21.8	13.5; 20.9	8.8; 17.1	4.3; 12.2	-0.7; 7.5	-5.3; 6.5
	-0.2	0.7	3.9	7.9	12.4	15.6	17.2	16.6	13.2	8.6	4.0	0.9
MV	-8.9; 5.1	-11.1; 6.0	-2.2; 7.0	3.8; 11.0	8.8; 15.8	11.4; 19.8	14.1; 21.9	13.8; 20.9	10.1; 17.6	5.0; 12.4	-0.5; 7.4	-5.5; 6.9
	-0.2	0.3	3.0	7.2	12.1	15.6	17.5	16.8	13.6	9.0	4.3	1.3
NI	-8.5; 6.0	-9.1; 7.0	-1.2; 7.8	4.4; 12.3	8.9; 16.6	11.0; 19.3	14.0; 22.1	13.7; 20.5	9.9; 17.6	4.9; 13.1	0.4; 8.6	-5.0; 8.3
	0.8	1.3	4.0	7.9	12.5	15.6	17.2	16.7	13.6	9.3	4.9	2.0
NI/HH/HB	-8.5; 6.0	-9.1; 6.9	-1.2; 7.8	4.4; 12.3	8.9; 16.6	11.0; 19.3	14.0; 22.1	13.7; 20.5	9.9; 17.6	4.9; 13.1	0.4; 8.6	-5.0; 8.3
	0.8	1.3	4.0	7.9	12.5	15.6	17.2	16.7	13.6	9.3	4.9	2.0
NW	-7.4; 6.0	-8.1; 7.5	-0.5; 8.3	4.3; 12.7	8.6; 16.3	11.0; 19.5	13.9; 22.3	13.9; 20.6	9.7; 17.8	5.1; 13.5	0.8; 8.8	-5.2; 8.4
	1.2	1.9	4.7	8.3	12.7	15.6	17.3	16.8	13.7	9.5	5.1	2.3
RP	-7.9; 5.0	-9.1; 7.0	-0.7; 8.1	4.4; 13.0	8.6; 16.0	11.5; 20.0	13.8; 22.6	13.6; 21.5	9.2; 17.7	4.8; 12.9	-0.3; 8.3	-5.0; 6.9
	0.3	1.2	4.4	8.2	12.6	15.8	17.5	16.8	13.5	9.0	4.3	1.4
SH	-7.5; 5.7	-8.8; 6.3	-1.3; 7.0	3.7; 10.8	8.7; 15.6	10.9; 18.9	14.0; 21.2	13.5; 20.8	10.4; 17.1	5.2; 12.8	0.6; 8.1	-3.8; 7.7
	0.7	0.9	3.2	7.1	11.7	15.1	16.9	16.4	13.5	9.2	4.9	2.0
SL	-7.3; 5.1	-8.7; 7.0	0.0; 8.4	4.6; 13.6	8.6; 16.0	11.6; 20.6	13.7; 23.2	13.7; 22.7	9.3; 18.0	4.9; 13.2	-0.4; 8.7	-4.7; 6.9
	0.6	1.6	4.8	8.5	12.8	16.0	17.7	17.0	13.7	9.2	4.5	1.6
SN	-10.3; 4.4	-11.5; 6.4	-3.0; 7.3	3.9; 12.8	8.8; 16.2	11.2; 20.9	14.4; 22.2	14.2; 21.0	8.7; 17.6	4.2; 12.4	-0.6; 7.7	-6.0; 6.3
	-0.9	-0.1	3.2	7.6	12.6	15.8	17.6	17.0	13.4	8.6	3.7	0.5
ST	-10.2; 5.4	-11.3; 7.0	-2.0; 7.7	4.8; 12.8	9.4; 17.2	11.5; 21.1	14.5; 22.9	14.5; 21.0	9.6; 17.8	4.7; 12.9	-0.2; 8.0	-5.6; 7.3
	0.0	0.7	3.8	8.2	13.1	16.3	18.0	17.4	13.9	9.1	4.4	1.3
TH/ST	-10.6; 4.8	-11.1; 6.6	-2.4; 7.3	4.3; 12.6	8.8; 16.4	11.1; 20.5	14.1; 22.3	14.1; 20.6	9.0; 17.3	4.4; 12.5	-0.7; 7.6	-5.9; 6.7
	-0.5	0.3	3.5	7.7	12.6	15.8	17.5	16.9	13.4	8.7	3.9	0.8
TH	-11.0; 4.1	-10.9; 6.2	-2.9; 6.8	3.7; 12.3	8.0; 15.5	10.5; 19.7	13.6; 21.4	13.4; 20.4	8.2; 16.8	3.7; 12.1	-1.2; 7.2	-6.2; 6.0
	-1.0	-0.2	3.1	7.2	12.0	15.2	16.8	16.2	12.8	8.2	3.7	0.2
Germany	-9.0; 4.8	-9.6; 6.6	-1.7; 7.5	4.3; 12.3	8.6; 16.0	11.2; 19.8	14.4; 22.0	14.0; 20.6	9.1; 17.3	4.5; 12.5	-0.2; 7.5	-5.3; 6.5
	-0.3	0.5	3.7	7.7	12.5	15.7	17.4	16.8	13.4	8.8	4.1	1.0

Note: Each table cell displays the minimum and maximum temperatures on the first line, followed by the monthly average temperature on the second line. BB: Brandenburg; BE: Berlin; BW: Baden-Wuerttemberg; BY: Bayern, HE: Hessen; MV: Mecklenburg-Vorpommern; NI: Niedersachsen; HH: Hamburg; HB: Bremen; NW: Nordrhein-Westfalen; RP: Rheinland-Pfalz; SH: Schleswig-Holstein; SL: Saarland; SN: Sachsen; ST: Sachsen-Anhalt; TH: Thueringen.

ARTICLE IN PRESS

ARTICLE IN PRESS

ARTICLE IN PRESS

To observe how the proposed algorithm works with multiplicative series, 10 time series data sets (Table 3) suitable for multiplicative series were randomly selected from the 'tsdl' package of R software. The R 'tsdl' package provides a collection of 648 time series datasets, primarily focusing on real-life series from various domains such as hydrology, finance, and production. The series vary in terms of their characteristics: some are seasonal (e.g., retail sales with yearly patterns), while others are non-seasonal (e.g., stock prices, unemployment rates). The number of data points in each series varies, typically ranging from a few dozen to several thousand observations, depending on the nature and frequency of the data. These series offer both short and long-term perspectives, making them useful for different types of time series analysis and modeling.

Table 3 Multiplicative time series data selected from the 'tsdl' package of R software

Dataset Abbreviation	Description
DS1	Monthly car sales in Quebec 1960-1968
DS2	Monthly U.S. male (16-19 years) unemployment figures (thousands) Jan. 1948-Dec. 1981
DS3	Current river at Van Buren, MO. Jan. 1922-Dec. 1960
DS4	Monthly production of chocolate confectionery in Australia: tonnes. Jul. 1957-Aug. 1995
DS5	Monthly production of sulphuric acid in Australia: thousand tonnes. Jan. 1956-Jun. 1994
DS6	Monthly flow, Bay du Nord River, Jan. 1953-Dec. 1981
DS7	Monthly river flow in cms, English River near Sioux Lookout, Jan. 1922-Dec. 1977
DS8	Monthly water usage (ml/day), London Ontario, Jan. 1966-Dec. 1988
DS9	The total generation of electricity by the U.S. electric industry (monthly data for the period Jan. 1985-Oct. 1996)
DS10	Monthly sales of new one-family houses sold in the USA Jan. 1973-Nov. 1995

ARTICLE IN PRESS

4.2 Comparison criteria

The Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Symmetric Mean Absolute Percentage Error (SMPE) introduced in Equations (30)–(33), are the commonly used evaluation metrics in literature for comparing the outcomes of the models in time series problems. In this study, the comparison relies on these metrics, which are described as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{30}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{31}$$

ARTICLE IN PRESS

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{32}$$

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \tag{33}$$

where y_i is the actual values of the time series and $\hat{y}_i$ is the forecasts.

When evaluating forecasting accuracy, different error metrics provide unique perspectives. RMSE emphasizes larger errors, while MAE treats all errors equally. MAPE expresses the average error as a percentage of the actual value, which can be helpful for understanding relative accuracy. SMAPE addresses a potential issue with MAPE by ensuring that overestimates and underestimates have the same impact on the calculated error. Each metric offers valuable insights, and the best choice often depends on the specific application and the relative importance of different types of errors.

4.3 System configuration

The implementation code was written in Python (version 3.7) on a PC (Intel Core i7-6700HQ, 2.6 GHz CPU, 20 GB RAM, 512 GB SSD) running Windows 10 operating system.

4.4 Experimental results for additive time series

To determine the forecasting performance of the proposed method on additive time series, the monthly average temperature data series of Germany and its states were used. The last 60 months of the 1728-month series were used as test data. The first 600 months of the series were designated as the training set, and the tracking error monitoring process was carried out within the time frame from the end of the training set to the end of the test set. To assess whether the proposed algorithm improves the performance of the WES method, the obtained performance values were compared with the performance values obtained from the WES functions of software packages such as SPSS, Python, and R. Furthermore, ablation studies were conducted to determine the individual contributions of each component within the proposed method, and the impact of training length on the model's performance was investigated. All results are presented in Sect. 4.4.1.

4.4.1 Additive time series forecasting results: Ablation studies and training length impact on the proposed method

Ablation studies were conducted to gain a deeper understanding of the individual contributions of each component within the proposed back-casting and GWO-based WES model with tracking error monitoring. By systematically removing or disabling components such as back-casting, GWO optimization, or the tracking error monitoring mechanism, the effect of each component on the overall forecasting accuracy was isolated. This approach allowed for the identification of the most critical components,

ARTICLE IN PRESS

the detection of potential redundancies, and a clear justification of the design choices made.

Furthermore, the impact of training length on the model's performance was investigated. This investigation aimed to determine the optimal amount of historical data needed for the model to achieve both accurate and robust forecasting performance. The inclusion of a tracking error monitoring mechanism in the proposed model made this analysis particularly relevant, as shorter training lengths could potentially enable faster adaptation to evolving data patterns, while longer training lengths might lead to overfitting or slower adaptation. By understanding the interplay between training length and the effectiveness of the tracking signal, the model's performance and resource usage could be optimized, especially in scenarios with limited historical data or dynamic time series with changing characteristics.

Table 4 presents the forecasting performance of the proposed method on monthly average temperature time series for Germany and its states with varying training lengths (600, 800, 1200, and 1500 months). While the proposed method demonstrates remarkable consistency across different training lengths, statistical tests reveal that shorter training lengths (600 months) tend to produce better results in terms of reducing the magnitude of errors (RMSE and MAE) and achieving lower percentage errors (MAPE and SMAPE). This observation is supported by Friedman tests, which indicate statistically significant differences in performance across the training lengths for all metrics (all p-values are less than 0.05). Nemenyi post-hoc tests further confirm that the performance values obtained with a 600-month training length are statistically different from those obtained with 1200 or 1500 months (all p-values are less than 0.05). Increasing the training length above 800 months does not have a significant effect on percentage errors. The effectiveness of the tracking error monitoring mechanism in enabling the model to adapt and generalize well, even with limited initial training data, likely contributes to the good performance observed with shorter training lengths.

To assess the individual contribution of tracking signal monitoring, the forecasting performance of the proposed WES method with and without tracking signal monitoring (as presented in Table 5), was compared to that of standard WES implementations in SPSS, Python, and R on monthly average temperature time series for Germany and its states using the Friedman tests. Results showed significant performance differences for each metric (RMSE, MAE, MAPE, and SMAPE) (p-values are all less than 0.001). This highlights the impact of method choice on forecasting accuracy. The proposed method, especially when incorporating tracking signal monitoring, consistently demonstrates superior performance across all datasets and most metrics. Nemenyi post-hoc tests revealed that the proposed method with tracking signal monitoring exhibits statistically significant differences ($p < 0.05$) compared to SPSS, Python, and R in terms of RMSE, MAPE, and SMAPE. The only exception was for MAE, where no significant difference was found between the proposed method with tracking signal monitoring and SPSS ($p = 0.191$). Furthermore, incorporating tracking signal monitoring led to significant performance improvements compared to the proposed method without monitoring for all metrics except SMAPE ($p = 0.152$). While generally performing well (lower average performance values against R and Python), the proposed

ARTICLE IN PRESS

Table 4 Forecasting performance values obtained on monthly average temperature time series for Germany and its states based on varying training lengths in the proposed method.

State	600 months				800 months				1200 months				1500 months			
	RMSE	**MAE**	**MAPE**	**SMAPE**	**RMSE**	**MAE**	**MAPE**	**SMAPE**	**RMSE**	**MAE**	**MAPE**	**SMAPE**	**RMSE**	**MAE**	**MAPE**	**SMAPE**
BB/BE	1.642	1.307	0.291	22.090	1.645	1.301	0.278	22.541	1.678	1.330	0.275	22.823	1.662	1.320	0.278	22.644
BB	1.644	1.309	0.297	22.204	1.645	1.302	0.282	22.662	1.678	1.330	0.279	22.943	1.663	1.322	0.282	22.765
BW	1.483	1.153	0.267	19.852	1.506	1.163	0.253	20.139	1.528	1.180	0.254	20.423	1.521	1.178	0.246	20.709
BY	1.509	1.134	0.239	25.761	1.539	1.159	0.238	28.268	1.559	1.176	0.235	28.213	1.550	1.171	0.238	28.691
HE	1.469	1.163	0.193	18.897	1.485	1.162	0.190	19.079	1.499	1.171	0.190	19.237	1.486	1.160	0.189	19.007
MV	1.493	1.170	0.227	20.340	1.508	1.178	0.223	20.829	1.546	1.199	0.220	21.098	1.537	1.194	0.222	20.950
NI	1.453	1.162	0.165	16.491	1.480	1.164	0.164	16.936	1.490	1.176	0.164	16.931	1.480	1.167	0.163	16.791
NI/HH/HB	1.455	1.162	0.165	16.486	1.470	1.160	0.163	16.763	1.490	1.177	0.164	16.942	1.477	1.166	0.163	16.746
NW	1.433	1.142	0.142	14.319	1.444	1.140	0.141	14.515	1.458	1.151	0.143	14.701	1.455	1.144	0.142	14.630
RP	1.465	1.162	0.161	15.970	1.485	1.173	0.161	16.412	1.501	1.180	0.162	16.547	1.493	1.175	0.161	16.402
SH	1.417	1.122	0.167	17.269	1.437	1.128	0.166	17.708	1.467	1.147	0.168	18.069	1.465	1.152	0.168	18.059
SL	1.476	1.202	0.159	15.590	1.494	1.204	0.157	15.725	1.497	1.204	0.157	15.690	1.492	1.198	0.154	15.420
SN	1.636	1.266	0.420	25.366	1.647	1.258	0.390	26.131	1.667	1.279	0.387	26.326	1.661	1.285	0.402	26.027
ST	1.566	1.241	0.237	20.166	1.579	1.239	0.230	20.554	1.609	1.272	0.230	20.981	1.597	1.261	0.232	20.711
TH/ST	1.550	1.217	0.297	22.206	1.570	1.226	0.285	22.849	1.590	1.243	0.283	23.071	1.578	1.234	0.283	22.795
TH	1.551	1.211	0.422	26.475	1.576	1.222	0.386	27.649	1.589	1.228	0.388	27.437	1.600	1.259	0.440	26.992
Germany	1.430	1.117	0.194	18.360	1.454	1.121	0.188	18.839	1.469	1.134	0.189	18.970	1.458	1.125	0.189	18.711
Avg.	1.510	1.190	0.238	19.873	1.527	1.194	0.229	20.447	1.548	1.210	0.229	20.612	1.540	1.206	0.232	20.473

ARTICLE IN PRESS

ARTICLE IN PRESS

method without monitoring doesn't consistently show statistically significant differences from the existing software packages (SPSS, Python, R). This suggests that while the core of the proposed method is sound, the tracking signal is crucial for achieving truly significant improvements. SPSS demonstrates statistically significantly better performance than R in terms of RMSE and MAE. This suggests that SPSS might be a preferable choice over R when minimizing these types of errors is a priority. SPSS also shows significantly better performance than Python for MAPE, indicating its potential advantage when accurate percentage error estimation is crucial. The use of the back-casting algorithm in SPSS, similar to the proposed method, and SPSS's generally better performance compared to R and Python, demonstrated the importance of the back-casting algorithm used in the proposed method within the established system.

4.4.2 Visualization of additive time series results and future forecasting

To visually determine the extent to which the forecasts obtained by the proposed method deviate from the actual time series data, a plot of the monthly average temperature data for Germany was selected as an example and presented in Figure 5.

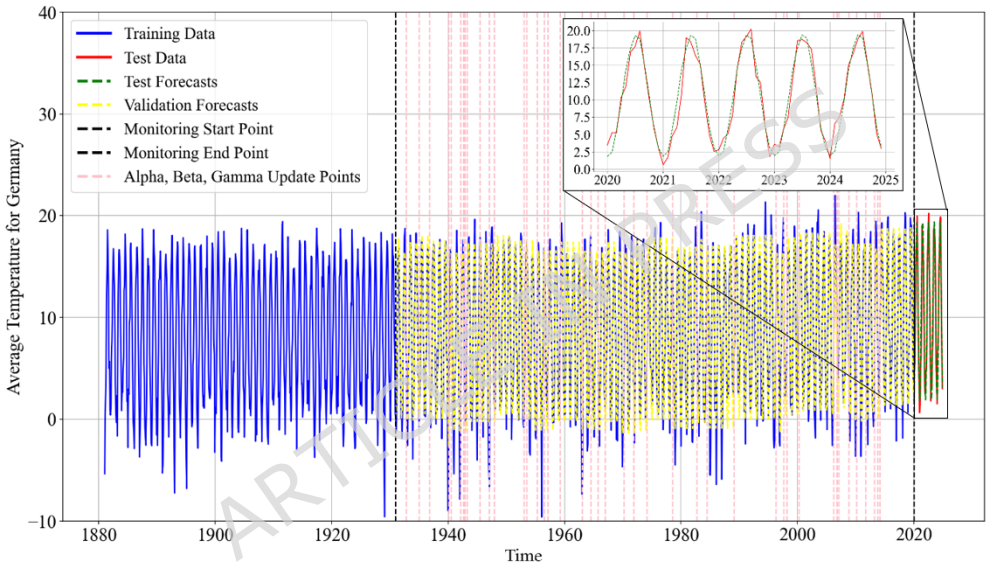


Fig. 5 Forecasting accuracy of the proposed method on the monthly average temperature data for Germany

Figure 5 illustrates the proposed method applied to the monthly average temperature time series for Germany, spanning the period from January 1881 to December 2024. It displays the training series, validation series, test series, validation forecasts, test forecasts, and the specific time points at which the alpha, beta, and gamma parameters of the WES method are updated during SETS monitoring. The inset plot provides a

ARTICLE IN PRESS

Table 5 Comparison of forecasting performance of different software packages on monthly average temperature time series for Germany and its states.

State	SPSS				Python				R				No TS				With TS			
	R	M	MP	SM	R	M	MP	SM	R	M	MP	SM	R	M	MP	SM	R	M	MP	SM
BB/BE	1.656	1.301	26.9	22.5	1.642	1.273	27.4	22.6	1.722	1.371	23.3	22.9	1.665	1.317	27.5	22.6	1.642	1.307	0.29	22.1
BB	1.656	1.302	27.3	22.7	1.642	1.274	27.6	22.7	1.721	1.371	23.4	23.0	1.665	1.318	27.9	22.7	1.644	1.309	0.30	22.2
BW	1.528	1.183	24.4	20.9	1.611	1.267	35.5	24.7	1.564	1.203	29.6	22.5	1.527	1.183	24.4	20.8	1.483	1.153	0.27	19.9
BY	1.552	1.176	24.1	30.1	1.591	1.227	99.8	35.3	1.559	1.172	49.0	28.9	1.552	1.175	24.1	29.6	1.509	1.134	0.24	25.8
HE	1.495	1.164	18.7	19.1	1.538	1.191	26.8	20.2	1.510	1.194	22.1	19.4	1.495	1.165	18.7	19.1	1.469	1.163	0.19	18.9
MV	1.543	1.195	22.0	21.1	1.541	1.194	25.3	21.0	1.562	1.232	22.2	20.9	1.545	1.198	22.1	21.0	1.493	1.170	0.23	20.3
NI	1.484	1.167	16.2	16.8	1.486	1.166	19.5	16.9	1.500	1.172	17.6	16.4	1.484	1.167	16.3	16.7	1.453	1.162	0.17	16.5
NI/HH/HB	1.484	1.167	16.2	16.7	1.486	1.166	19.5	16.9	1.500	1.172	17.6	16.4	1.484	1.167	16.2	16.7	1.455	1.162	0.17	16.5
NW	1.454	1.143	14.1	14.6	1.532	1.204	19.1	16.1	1.472	1.163	15.6	14.6	1.454	1.143	14.1	14.6	1.433	1.142	0.14	14.3
RP	1.496	1.177	16.0	16.5	1.560	1.224	21.7	17.9	1.526	1.223	19.7	17.3	1.496	1.177	16.0	16.4	1.465	1.162	0.16	16.0
SH	1.471	1.158	16.9	18.2	1.466	1.150	21.7	17.9	1.436	1.131	19.4	17.2	1.470	1.157	16.9	18.1	1.417	1.122	0.17	17.3
SL	1.494	1.197	15.3	15.4	1.577	1.221	19.6	16.7	1.561	1.256	19.6	17.0	1.493	1.197	15.4	15.4	1.476	1.202	0.16	15.6
SN	1.640	1.249	37.8	26.0	1.627	1.223	37.8	27.2	1.711	1.322	27.9	26.8	1.647	1.262	38.6	26.0	1.636	1.266	0.42	25.4
ST	1.598	1.257	22.7	20.7	1.594	1.242	24.9	20.9	1.634	1.293	21.6	20.9	1.601	1.261	22.9	20.7	1.566	1.241	0.24	20.2
TH/ST	1.578	1.231	28.0	22.8	1.583	1.220	30.1	23.4	1.612	1.261	24.4	23.1	1.580	1.234	28.2	22.7	1.550	1.217	0.30	22.2
TH	1.572	1.217	38.8	27.2	1.615	1.254	52.2	30.5	1.604	1.245	31.0	27.7	1.574	1.219	39.2	27.2	1.551	1.211	0.42	26.5
Germany	1.464	1.127	18.6	18.8	1.483	1.128	25.9	19.6	1.479	1.148	21.3	19.0	1.464	1.128	18.7	18.8	1.430	1.117	0.19	18.4
Avg.	1.539	1.201	22.6	20.6	1.563	1.213	31.4	21.8	1.569	1.231	23.8	20.8	1.541	1.204	22.8	20.5	1.510	1.190	0.24	19.9

Legend: R = RMSE, M = MAE, MP = MAPE, SM = SMAPE, No TS = Proposed Method (No Tracking Signal), With TS = Proposed Method (With Tracking Signal)

ARTICLE IN PRESS

ARTICLE IN PRESS

closer look at the model's accuracy during the test period. The inset plot highlights the close agreement between the forecasts (green dashed line) and the actual data (red line).

According to Figure 5, the model accurately captures the overall trend and seasonality of the temperature data. The validation forecasts show that the model was able to follow the data during the training period. The frequent update points for alpha, beta, and gamma suggest a dynamic model that adapts to changes. The inset graph shows that the model is very close to the test data.

Utilizing the devised method, future average temperatures across Germany were projected, focusing on estimations for 2025-2029. The findings, displayed in Table 6, exhibit a consistent pattern of temperature change across the months and a gradual year-to-year increase. This gradual warming trend can be interpreted as a potential indicator of climate change.

Table 6 Monthly average temperature values forecasted for Germany from 2025 to 2029 using the proposed method.

Year	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
2025	2.047	2.641	5.826	9.876	14.453	17.574	19.483	18.981	15.924	10.571	6.111	3.198
2026	2.040	2.647	5.859	9.896	14.468	17.607	19.471	19.047	15.298	10.571	6.153	3.232
2027	2.013	2.656	5.799	9.855	14.436	17.577	19.455	19.012	15.213	10.466	6.138	3.210
2028	2.079	2.645	5.799	9.885	14.573	17.635	19.552	19.028	15.355	10.520	6.183	3.245
2029	2.140	2.735	5.893	9.923	14.707	17.701	19.770	19.141	15.370	11.000	6.244	3.430

ARTICLE IN PRESS

4.5 Experimental results for multiplicative time series

To assess the forecasting performance of the proposed method on multiplicative time series, ten time series data from Table 3 were utilized. 60% of these series were used as training data, and 10% were used as test data. The tracking error monitoring process was applied in the time period between the training set and the test set. An assessment of the proposed algorithm's efficacy in enhancing WES method performance was conducted through a comparison of obtained performance values with those derived from the WES functions of software packages including SPSS, Python, and R. Furthermore, ablation studies were undertaken to ascertain the individual contributions of each component within the proposed method, and the impact of training length on the model's performance was examined. The results are comprehensively presented in Sect. 4.5.1.

4.5.1 Multiplicative time series forecasting results: ablation studies and training length impact on the proposed method

Table 7 presents the forecasting performance of the proposed method across 10 different datasets (DS1 to DS10) with varying training lengths (ranging from 20% to 70% of the time series length, denoted as 0.2×n to 0.7×n). The table demonstrates that the proposed method's performance varies significantly across different multiplicative

ARTICLE IN PRESS

time series. This is expected, as each series likely has unique characteristics (e.g., seasonality, trend, noise). In some time series (DS1, DS3, DS10), there is a general trend of improvement in RMSE and MAE as the training length increases, while for other time series (DS4, DS5), the trend is less consistent or even reversed. The behavior of MAPE and SMAPE is more complex and varies across datasets. In some cases, performance improves with longer training lengths, while in others, it fluctuates.

When the performance values obtained using the WES function in SPSS, Python, and R software packages in Table 8 were compared with the performance values obtained by the proposed method for varying training lengths, it was observed that, despite changes in the training length, the proposed method provided better performance values in most cases.

To assess the individual contribution of the tracking signal monitoring, the forecasting performance of the WES method in SPSS, Python, and R (Table 8) was compared to that of the proposed method with and without tracking signal monitoring using the Friedman test. The results indicate that the performances differ significantly for each metric (RMSE, MAE, MAPE, and SMAPE) (p-values are <0.001, 0.009, 0.007, and 0.003, respectively). This demonstrates that the choice of forecasting technique significantly impacts performance. The proposed method, enhanced with tracking signal monitoring, consistently exhibited notably lower values across all metrics compared to the other methods (Nemenyi post hoc tests, all p-values are less than 0.05). The WES functions within SPSS, Python, and R showed no statistically significant performance differences among themselves. Notably, R tended to underperform, particularly in percentage-based error measures (MAPE and SMAPE). Integrating tracking signal monitoring into the proposed method resulted in a substantial improvement over the proposed method. Even without this enhancement, the proposed method generally outperformed the WES functions of SPSS, Python, and R, although these differences were not always statistically significant.

5 Robustness analysis and comprehensive benchmark

5.1 Rationale for extended evaluation

The preliminary evaluations presented in the previous section demonstrate the efficacy of the proposed framework on specific climate and industry datasets. However, to ensure that the performance gains are not confined to a single geographic or temporal context, a more rigorous validation is required. This section is introduced to address critical methodological considerations, such as potential sampling bias in fixed data splits, the generalizability of the model across diverse frequencies, and the stability of the metaheuristic optimization process. Furthermore, an investigation into one-step-ahead forecasting is included to demonstrate the real-time adaptive capabilities of the proposed framework.

Table 7 Forecasting results for multiplicative time series with varying training lengths

	Training Length							Training Length					
Dataset	0.2×n	0.3×n	0.4×n	0.5×n	0.6×n	0.7×n	**Dataset**	0.2×n	0.3×n	0.4×n	0.5×n	0.6×n	0.7×n
DS1							**DS6**						
RMSE	2600.095	2336.569	1821.717	1872.154	1773.478	1826.378	RMSE	21.912	21.325	19.767	20.566	20.005	20.609
MAE	2025.798	1827.876	1559.586	1588.603	1508.709	1567.074	MAE	17.346	16.526	14.804	15.355	14.833	15.562
MAPE	10.720	9.785	8.341	8.507	8.062	8.424	MAPE	47.342	45.927	42.085	43.745	42.621	44.052
SMAPE	11.669	10.581	8.581	8.919	8.414	8.649	SMAPE	46.969	43.750	38.480	39.694	38.458	40.594
DS2							**DS7**						
RMSE	75.062	68.424	82.033	206.768	94.741	98.694	RMSE	96.019	81.791	81.201	84.700	83.700	80.304
MAE	59.964	55.237	65.029	158.811	74.168	77.246	MAE	70.635	53.078	54.425	62.576	61.669	56.629
MAPE	6.554	6.155	7.132	17.065	6.052	8.376	MAPE	87.762	49.396	54.483	82.773	83.027	71.508
SMAPE	6.800	6.218	7.427	13.333	5.823	8.908	SMAPE	50.379	41.565	42.479	46.156	45.698	42.462
DS3							**DS8**						
RMSE	59.584	54.775	50.604	49.386	49.829	49.499	RMSE	17.210	18.040	17.904	18.104	18.124	18.638
MAE	41.840	36.650	30.083	28.096	28.453	28.346	MAE	11.908	12.967	13.779	13.543	13.320	13.355
MAPE	70.401	65.098	43.538	37.506	38.236	39.926	MAPE	7.564	7.207	7.606	7.389	7.411	7.895
SMAPE	112.303	91.377	59.623	53.468	53.934	53.630	SMAPE	6.898	7.660	8.264	8.071	8.171	7.895
DS4							**DS9**						
RMSE	1120.517	1093.930	1178.161	1239.534	1247.803	1251.271	RMSE	7.806	7.775	7.062	6.889	6.472	5.454
MAE	817.418	777.291	855.481	917.724	928.358	936.058	MAE	5.868	6.342	6.013	7.764	5.255	4.549
MAPE	9.851	9.406	10.176	10.797	10.911	10.982	MAPE	2.246	2.467	2.349	3.054	2.027	1.738
SMAPE	10.277	10.108	11.156	11.925	12.067	12.149	SMAPE	2.247	2.449	2.329	3.012	2.022	1.732
DS5							**DS10**						
RMSE	43.552	44.972	47.604	53.985	51.308	55.501	RMSE	6.299	6.161	5.815	5.404	5.172	4.729
MAE	42.057	43.398	45.186	52.744	49.637	54.357	MAE	5.179	5.290	4.807	3.865	3.599	3.237
MAPE	60.598	62.570	66.409	75.861	71.131	73.880	MAPE	9.024	9.080	8.707	3.545	6.621	6.238
SMAPE	45.518	46.641	48.855	54.016	51.433	55.482	SMAPE	9.506	9.587	8.975	8.619	6.935	6.501

Note: n is the length of the corresponding time series

ARTICLE IN PRESS

ARTICLE IN PRESS

Table 8 Comparison of forecasting performance of different software packages on multiplicative time series

	Metric	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8	DS9	DS10
SPSS	RMSE	1910.1	104.5	50.8	1264.4	58.2	20.4	88.2	19.5	8.2	5.9
	MAE	1560.5	81.0	30.1	950.5	57.1	15.3	68.3	14.9	5.2	4.8
	MAPE	8.32	8.75	43.34	11.06	82.58	46.65	92.41	8.38	1.94	8.44
	SMAPE	8.63	9.38	59.30	12.16	57.28	38.68	50.06	8.93	1.99	8.95
Python	RMSE	1893.7	106.6	50.1	1372.1	60.8	20.8	378.2	17.1	7.95	6.57
	MAE	1492.3	82.7	28.7	1068.6	59.5	15.9	267.4	11.9	5.18	5.41
	MAPE	8.56	10.39	98.02	12.21	45.13	53.19	67.21	7.46	2.02	10.85
	SMAPE	8.17	9.59	54.28	12.18	59.02	42.27	95.56	6.98	1.98	10.07
R	RMSE	1899.8	153.1	50.9	1349.8	60.4	21.5	112.7	19.0	9.24	19.0
	MAE	1487.0	132.2	29.6	1043.6	58.4	16.8	85.5	13.9	6.88	17.7
	MAPE	7.93	14.79	42.00	12.00	83.49	47.80	117.9	7.72	2.00	31.61
	SMAPE	8.35	16.34	57.74	13.31	57.70	43.88	57.57	8.22	2.66	38.55
Proposed (No TS)	RMSE	1837.1	106.6	50.7	1250.7	52.6	20.6	89.5	18.7	8.16	5.72
	MAE	1565.8	82.6	29.5	932.9	51.0	15.6	69.1	13.4	6.09	6.07
	MAPE	8.38	8.94	41.59	10.97	75.17	43.11	94.10	7.43	2.69	8.53
	SMAPE	8.61	9.58	56.68	12.15	52.97	41.21	50.84	7.91	2.66	8.50
Proposed (With TS)	RMSE	1773.5	94.7	49.8	1247.8	51.3	20.0	83.7	18.6	6.47	5.17
	MAE	1508.7	74.2	28.5	928.4	49.6	14.8	61.7	13.3	5.26	3.87
	MAPE	8.06	8.05	39.83	10.91	71.13	42.62	82.04	7.39	2.03	6.62
	SMAPE	8.41	8.52	53.93	12.07	51.43	38.46	45.70	7.87	2.02	6.94

Legend: TS = Tracking Signal, DS(i) = Dataset (i)

ARTICLE IN PRESS

ARTICLE IN PRESS

ARTICLE IN PRESS

5.2 Addressing dataset diversity (M4 competition)

To establish the external validity of the proposed method, the experimental scope is expanded to include 40 diverse series from the M4 Forecasting Competition (Makridakis et al. 2019). These datasets are selected to represent four distinct temporal frequencies: monthly, weekly, hourly, and quarterly. By utilizing series with varying trend slopes, seasonal strengths, and noise levels, the generalizability of the proposed method is systematically tested. This ensures that the adaptive bias adjustment remains effective regardless of the underlying data source or the domain (e.g., finance, industry, or social demographics).

5.3 Rolling-origin evaluation framework

The limitations of fixed data splitting (e.g., a single 60–30–10 split) are mitigated by employing a rolling-origin evaluation protocol. Instead of a single performance estimate, the model is evaluated across four sequential forecast origins (folds). In each fold, the forecast origin is shifted forward by a duration equal to the forecast horizon (h). This approach provides a statistically robust estimate of forecasting accuracy by accounting for the temporal evolution of the series, thereby eliminating the risk of sampling bias associated with static validation windows.

To assess the statistical significance of the performance differences observed among the evaluated models, a non-parametric Friedman test was conducted on the ranking data across the 40 time series (Table 9, Figure 6). The null hypothesis, which assumes that all forecasting methods perform equally, was rigorously rejected with a Friedman statistic of $\chi^2 = 66.8822$ and a highly significant p-value (< 0.001). This result confirms that the differences in forecasting accuracy across the proposed method, Static Holt-Winters, Auto ARIMA, ETS, and Seasonal Naive are not due to random chance. Following the rejection of the null hypothesis, a Post-hoc Nemenyi test was performed to identify specific pairwise differences between the methods. The results of the Nemenyi test demonstrated that the Proposed Method significantly outperformed all other benchmarks at the $\alpha = 0.05$ significance level. Specifically, the Proposed Method established a clear statistical advantage over the Static Holt-Winters model ($p = 0.0341$) and demonstrated superior performance compared to Auto ARIMA ($p = 0.0010$), ETS ($p = 0.0010$), and Seasonal Naive ($p = 0.0010$). The stability and robustness of the proposed framework were further highlighted by the mean rank analysis. The proposed method secured the top position with a mean rank of 1.475, being significantly closer to the ideal rank of 1.0 compared to its closest rival, Static Holt-Winters (rank: 2.500). Classical statistical models like ETS (3.250), Seasonal Naive (3.875), and Auto ARIMA (3.900) followed with higher ranks. These findings statistically validate that the integration of GWO-based parameter optimization and SETS-driven adaptive bias correction provides a substantial and consistent improvement in forecasting accuracy over conventional methodologies.

5.4 Systematic GWO hyperparameter stability analysis

The stability of GWO is scrutinized through a systematic hyperparameter sensitivity analysis. Rather than selecting the population size and iteration count based on

ARTICLE IN PRESS

Table 9 SMAPE values and performance ranks of the proposed and benchmark methods across 40 M4 competition series

Group	ID	Proposed Method	Holt-Winters	Auto ARIMA	ETS	Seasonal Naive
Quarterly	Q1024	1.4052 (1)	1.4052 (2)	1.9778 (4)	1.7799 (3)	3.5116 (5)
Quarterly	Q1100	6.9400 (1)	7.3652 (2)	7.4263 (3)	8.7955 (5)	7.8353 (4)
Quarterly	Q10210	5.0788 (2)	4.9496 (1)	8.8019 (5)	7.4254 (3)	7.4703 (4)
Quarterly	Q10446	3.4542 (1)	3.5814 (3)	4.4271 (5)	3.9013 (4)	3.5321 (2)
Quarterly	Q10499	0.8094 (1)	0.8095 (2)	1.3943 (4)	1.2841 (3)	2.6986 (5)
Quarterly	Q10580	2.2232 (1)	2.2401 (2)	2.8286 (3)	3.1260 (5)	2.8983 (4)
Quarterly	Q10583	3.5365 (1)	3.5745 (2)	7.2138 (5)	5.8856 (4)	3.9240 (3)
Quarterly	Q10592	5.5274 (1)	7.3654 (2)	16.4798 (4)	16.4721 (3)	26.7119 (5)
Quarterly	Q10608	3.2025 (1)	3.3727 (2)	5.2623 (5)	4.2757 (3)	4.3622 (4)
Quarterly	Q10612	1.9399 (1)	4.3472 (5)	3.4417 (3)	2.6688 (2)	3.8891 (4)
Monthly	M17	0.8822 (1)	0.8822 (2)	1.1804 (5)	0.9618 (4)	0.9478 (3)
Monthly	M19	11.4096 (1)	11.6405 (2)	13.4439 (3)	15.1715 (5)	13.9453 (4)
Monthly	M22	19.4500 (2)	20.8655 (3)	26.5128 (5)	25.7501 (4)	18.1649 (1)
Monthly	M23	9.7752 (1)	9.8438 (2)	12.8337 (5)	10.5577 (4)	10.4545 (3)
Monthly	M133	13.5783 (2)	13.8745 (3)	14.6356 (4)	12.8129 (1)	18.1250 (5)
Monthly	M135	1.4302 (1)	1.4316 (2)	2.3910 (5)	1.6837 (3)	1.7545 (4)
Monthly	M142	11.3760 (2)	11.3328 (1)	17.2969 (4)	16.7566 (3)	22.0213 (5)
Monthly	M151	9.7325 (2)	9.9833 (3)	10.7055 (4)	9.6230 (1)	13.4509 (5)
Monthly	M188	5.1873 (2)	5.3991 (4)	5.2170 (3)	4.7807 (1)	9.0198 (5)
Monthly	M191	6.1090 (2)	6.8799 (3)	6.0871 (1)	8.6887 (4)	9.0713 (5)
Weekly	W2	3.2627 (1)	3.6266 (2)	4.7978 (3)	4.8841 (4)	5.8041 (5)
Weekly	W19	11.2083 (1)	16.4804 (4)	21.0745 (5)	13.0223 (2)	13.5096 (3)
Weekly	W57	8.9533 (2)	10.8623 (3)	10.9121 (4)	7.1993 (1)	16.5550 (5)
Weekly	W74	12.4328 (1)	12.4635 (2)	26.7050 (5)	19.0241 (3)	19.3814 (4)
Weekly	W83	4.5051 (2)	4.9311 (4)	4.5367 (3)	4.2987 (1)	18.7092 (5)
Weekly	W96	0.7155 (2)	0.7155 (2)	1.1096 (4)	0.6959 (1)	4.2771 (5)
Weekly	W110	1.9190 (2)	2.0084 (3)	1.5834 (1)	2.3286 (4)	8.4517 (5)
Weekly	W116	4.1783 (1)	4.8957 (4)	10.3251 (5)	4.8334 (3)	4.6123 (2)
Weekly	W235	9.0548 (1)	9.3623 (2)	32.6021 (4)	34.7512 (5)	17.1983 (3)
Weekly	W239	10.7975 (2)	10.7898 (1)	14.6509 (4)	29.2964 (5)	13.3721 (3)
Hourly	H10	3.9719 (2)	4.3214 (4)	5.6051 (5)	4.1024 (3)	3.3281 (1)
Hourly	H31	4.4437 (2)	5.8175 (4)	11.9749 (5)	2.7280 (1)	4.4706 (3)
Hourly	H55	4.9780 (1)	5.5528 (2)	7.5828 (5)	6.6511 (4)	5.7548 (3)
Hourly	H73	15.0178 (1)	15.0824 (2)	19.8648 (4)	28.2401 (5)	15.3527 (3)
Hourly	H105	11.6116 (2)	11.8402 (3)	10.8661 (1)	15.5741 (5)	13.5444 (4)
Hourly	H133	36.9960 (2)	36.4440 (1)	62.6139 (5)	58.8180 (4)	41.3911 (3)
Hourly	H311	0.5265 (3)	0.5687 (4)	0.4624 (1)	0.4712 (2)	1.1821 (5)
Hourly	H318	0.2557 (1)	0.2571 (2)	0.3279 (4)	0.2919 (3)	0.7594 (5)
Hourly	H401	20.6617 (2)	20.3462 (1)	35.7352 (5)	27.7077 (3)	28.2577 (4)
Hourly	H410	25.4369 (1)	25.9450 (2)	31.1519 (3)	35.9986 (5)	32.7171 (4)
Mean Rank		1.475	2.5	3.9	3.25	3.875

heuristic assumptions, a grid-search analysis is performed across configurations where the population size in $5, 15, 30$ and iteration count in $10, 25, 50$. The performance is measured not only by the mean SMAPE but also by the standard deviation across multiple runs. This systematic tuning ensures that the optimization process is deep enough to reach convergence while maintaining stability and computational efficiency.

ARTICLE IN PRESS

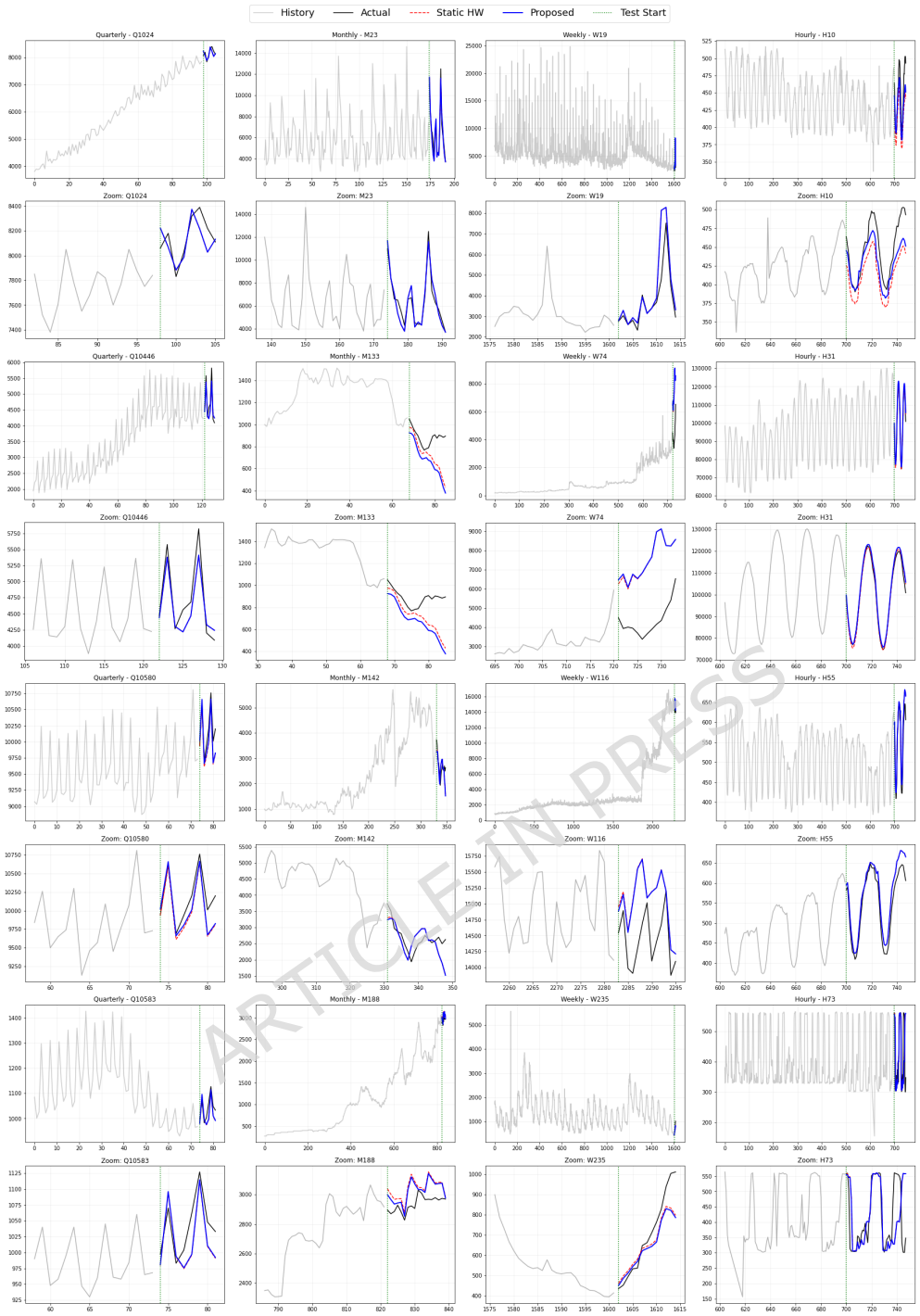


Fig. 6 Representative multi-step out-of-sample forecasting results for Quarterly, Monthly, Weekly, and Hourly series, comparing the proposed SETS-HW method (blue) against the static HW model (red dashed)

ARTICLE IN PRESS

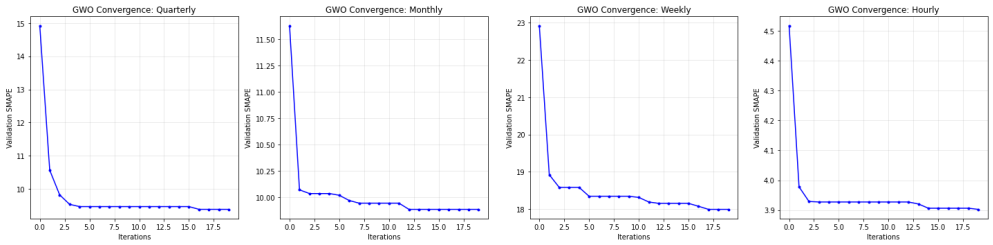


Fig. 7 GWO convergence curves showing validation SMAPE reduction across representative series from different temporal frequencies

5.5 Convergence and computational complexity analysis

The efficiency of the optimization layer is documented via GWO convergence behavior plots (Figure 7). These plots demonstrate how the SMAPE decreases as the search agents navigate the parameter space, typically reaching a stable minimum within the first 15–20 iterations.

Figure 7 illustrates the GWO convergence curves across representative series from each frequency group (quarterly, monthly, weekly, and hourly) derived from the M4 Forecasting Competition dataset. Technically, these convergence profiles are obtained by monitoring the best fitness value (Alpha wolf's validation SMAPE) during the search process, where each agent in the GWO population represents a candidate configuration for the smoothing parameters (α, β, γ) and the monitoring threshold. The sharp declines in the plots correspond to instances where the swarm discovers superior parameter regions (exploration), while the plateauing segments indicate the fine-tuning of the current best solution (exploitation). A remarkably consistent convergence pattern is observed across almost all datasets regardless of their frequency or volatility, demonstrating the stability of the optimization layer. Ultimately, this rapid and uniform decline in SMAPE across diverse groups confirms the algorithm's high exploration efficiency and its capacity to provide robust parameter configurations within a computationally compact window.

Additionally, a computational complexity analysis is provided to compare the runtime of the proposed framework against established benchmarks such as Auto-ARIMA and ETS. This discussion highlights the trade-off between the computational cost of metaheuristic calibration and the resulting gains in forecasting accuracy.

As shown in Table 10, the computational efficiency of the framework offers a compelling advantage for real-time deployment. As detailed in the complexity summary, while the metaheuristic calibration (GWO) introduces a training overhead, the normalized time reveals that the proposed method requires only 1.58 ms per observation. In contrast, Auto-ARIMA requires approximately 82.85 ms per observation, making it nearly 52 times more computationally expensive without matching the adaptive accuracy of the proposed model. The median raw time of 533.28 ms for the proposed method confirms that even with swarm-based optimization, the framework can process

ARTICLE IN PRESS

Table 10 Computational efficiency comparison of the proposed and benchmark methods across 40 M4 competition series

Methods	Average Raw Time (ms)	Normalized Time (ms/obs)	Median Raw Time (ms)
Proposed Method	553.8364	1.5840	533.2844
Static Holt-Winters	2.6672	0.0076	2.5010
Seasonal Naive	0.0140	0.0001	0.0146
Auto ARIMA	54268.5501	82.8520	29029.2389
ETS	182.4982	0.4575	172.9514

Note: Normalized time divides total execution by number of data points to ensure fair comparison.

a full series and generate optimized forecasts in approximately half a second. Furthermore, the SETS monitoring phase adds negligible overhead ($O(n)$) to the standard Holt-Winters update logic, supporting its suitability for large-scale data processing and high-frequency environments.

5.6 Real-time adaptability via one-step-ahead forecasting

To demonstrate the model's suitability for streaming data and real-time deployment, a one-step-ahead forecast analysis is conducted. In this sequential prediction framework, the model generates a forecast for $(t + 1)$, and upon observing the actual value, the SETS tracking signal immediately evaluates the forecast error. This allows for an instantaneous update of the bias adjustment term if a structural break or level shift is detected. The effectiveness of the method in this online setting provides strong evidence for its practical applicability in high-frequency domains, where maintaining control over cumulative error is paramount.

To rigorously evaluate the statistical significance of the forecasting improvements in the one-step-ahead scenario, a non-parametric Friedman test was conducted on the performance ranks across the 40 test series (Table 11, Figure 8). The null hypothesis, which assumes that all forecasting models perform equally, was decisively rejected with a Friedman statistic of $\chi^2 = 68.4266$ and a highly significant p-value (< 0.001). This result provides strong evidence that the performance variations among the evaluated methods are statistically meaningful and not due to random chance.

Following the Friedman test, a Post-hoc Nemenyi test was performed to identify specific pairwise differences and performance groupings. The analysis revealed that the Proposed Method (Mean Rank = 1.625) and the Static Holt-Winters model (Mean Rank = 2.200) form the top-performing cluster. While the Proposed Method achieved a superior mean rank, the difference between these two specific models was not statistically significant at the $\alpha = 0.05$ level ($p = 0.4593$). However, the Proposed Method demonstrated highly significant superiority ($p = 0.0010$) over all other classical benchmarks, including Auto ARIMA (3.600), ETS (3.600), and Seasonal Naive (4.000).

The ranking distribution highlights the robustness of the proposed framework. By securing a mean rank of 1.625, the Proposed Method consistently positioned itself at the top of the hierarchy, effectively bridging the gap toward the ideal rank of 1.0. These

ARTICLE IN PRESS

findings validate that while the back-casting initialization and GWO optimization provide a strong baseline (shared with the static version), the integration of the SETS error-tracking mechanism ensures the most stable performance, particularly in outperforming established automated frameworks like ARIMA and ETS by a substantial statistical margin.

Table 11 SMAPE values and performance ranks of the proposed and benchmark methods across 40 M4 competition series (One-step-ahead forecasting)

Group	ID	Proposed Method	Holt-Winters	Auto ARIMA	ETS	Seasonal Naive
Quarterly	Q1024	1.4315 (2)	1.4377 (3)	1.5954 (4)	1.2877 (1)	3.2651 (5)
Quarterly	Q1100	2.8128 (1)	2.9194 (2)	3.7112 (5)	3.2882 (4)	3.1618 (3)
Quarterly	Q10210	2.0292 (2)	2.0073 (1)	8.1128 (5)	4.8476 (3)	7.0607 (4)
Quarterly	Q10446	4.3043 (5)	4.1719 (4)	3.6411 (1)	4.0420 (3)	3.9454 (2)
Quarterly	Q10499	0.2839 (2)	0.2844 (3)	0.2902 (4)	0.2664 (1)	3.8585 (5)
Quarterly	Q10580	0.6628 (1)	0.7025 (2)	2.0562 (4)	2.1398 (5)	1.8532 (3)
Quarterly	Q10583	1.7968 (1)	2.1569 (2)	3.2661 (3)	5.0704 (5)	4.5307 (4)
Quarterly	Q10592	5.1557 (1)	6.3209 (2)	9.0168 (3)	9.1068 (4)	24.4888 (5)
Quarterly	Q10608	2.1422 (2)	2.7661 (4)	2.3609 (3)	1.5970 (1)	4.6186 (5)
Quarterly	Q10612	1.8389 (2)	1.8684 (3)	2.0059 (4)	1.2031 (1)	3.5534 (5)
Monthly	M17	0.3400 (1)	0.3396 (1)	0.3626 (3)	0.4738 (5)	0.4590 (4)
Monthly	M19	7.6245 (2)	6.4069 (1)	13.4665 (4)	9.4157 (3)	13.6618 (5)
Monthly	M22	13.3704 (3)	20.0693 (5)	12.9314 (1)	13.1844 (2)	18.6723 (4)
Monthly	M23	8.5384 (1)	9.6093 (3)	8.8150 (2)	14.4828 (5)	9.8195 (4)
Monthly	M133	5.3057 (1)	6.1806 (2)	22.4774 (4)	8.7881 (3)	30.2004 (5)
Monthly	M135	1.0056 (1)	1.5571 (2)	2.2580 (4)	2.1519 (3)	3.2148 (5)
Monthly	M142	9.4875 (1)	9.7506 (2)	30.2229 (3)	31.3643 (4)	35.2655 (5)
Monthly	M151	5.6392 (2)	5.1977 (1)	13.6000 (3)	15.9565 (4)	18.9522 (5)
Monthly	M188	1.8126 (2)	1.8558 (3)	1.6872 (1)	3.3507 (5)	2.1791 (4)
Monthly	M191	0.6923 (1)	0.7239 (2)	2.7953 (4)	3.2100 (5)	2.3405 (3)
Weekly	W2	1.9518 (2)	1.8784 (1)	3.8079 (4)	3.2442 (3)	9.6410 (5)
Weekly	W19	9.7285 (1)	10.7075 (4)	20.7955 (5)	10.6080 (3)	10.4136 (2)
Weekly	W57	8.0144 (2)	8.3400 (3)	17.6384 (5)	11.3972 (4)	5.1975 (1)
Weekly	W74	8.2307 (2)	7.9905 (1)	20.9423 (4)	14.1435 (3)	27.1756 (5)
Weekly	W83	0.6385 (1)	0.6409 (2)	1.2426 (3)	3.2821 (4)	25.9042 (5)
Weekly	W96	0.4229 (1)	0.4255 (2)	1.7717 (4)	0.7492 (3)	4.1473 (5)
Weekly	W110	0.7960 (2)	0.9055 (3)	0.5527 (1)	1.2080 (4)	14.1509 (5)
Weekly	W116	2.6382 (1)	2.7584 (2)	10.3229 (5)	6.0347 (4)	2.9973 (3)
Weekly	W235	14.9716 (1)	20.6165 (3)	59.5222 (5)	46.9462 (4)	18.0062 (2)
Weekly	W239	4.8222 (1)	6.6480 (2)	16.4671 (4)	33.5817 (5)	14.8159 (3)
Hourly	H10	1.2840 (1)	1.4833 (2)	2.2233 (4)	2.1396 (3)	2.6568 (5)
Hourly	H31	2.6787 (1)	4.9970 (2)	11.1353 (4)	105.3682 (5)	6.4420 (3)
Hourly	H55	1.4851 (1)	1.7750 (2)	4.5444 (5)	3.5044 (3)	4.2170 (4)
Hourly	H73	9.3067 (2)	8.7135 (1)	16.1638 (4)	27.8525 (5)	15.3217 (3)
Hourly	H105	3.4976 (1)	3.5515 (2)	6.7256 (4)	13.7550 (5)	5.6555 (3)
Hourly	H133	32.5619 (3)	28.0225 (2)	42.7385 (4)	58.6598 (5)	22.6369 (1)
Hourly	H311	0.2641 (2)	0.2614 (1)	0.3025 (3)	0.4112 (4)	0.6801 (5)
Hourly	H318	0.1481 (2)	0.1428 (1)	0.2293 (3)	0.2303 (4)	0.7955 (5)
Hourly	H401	24.3229 (2)	29.3440 (4)	34.4193 (5)	20.9104 (1)	25.1909 (3)
Hourly	H410	24.2086 (2)	22.8105 (1)	30.6692 (3)	33.5203 (5)	32.8854 (4)
Mean Rank		1.625	2.2	3.6	3.6	4.0

ARTICLE IN PRESS

5.7 Practical implementation: Data-driven threshold selection

A key feature of the practical implementation is the automated selection of the monitoring threshold. Rather than being manually assigned, the threshold is treated as a decision variable within the GWO optimization phase. This ensures that the sensitivity of the error tracking signal is automatically calibrated to the specific volatility of each

ARTICLE IN PRESS

dataset. Consequently, the framework remains highly responsive to structural breaks in volatile environments while preventing excessive model updates in stable scenarios.

ARTICLE IN PRESS

ARTICLE IN PRESS

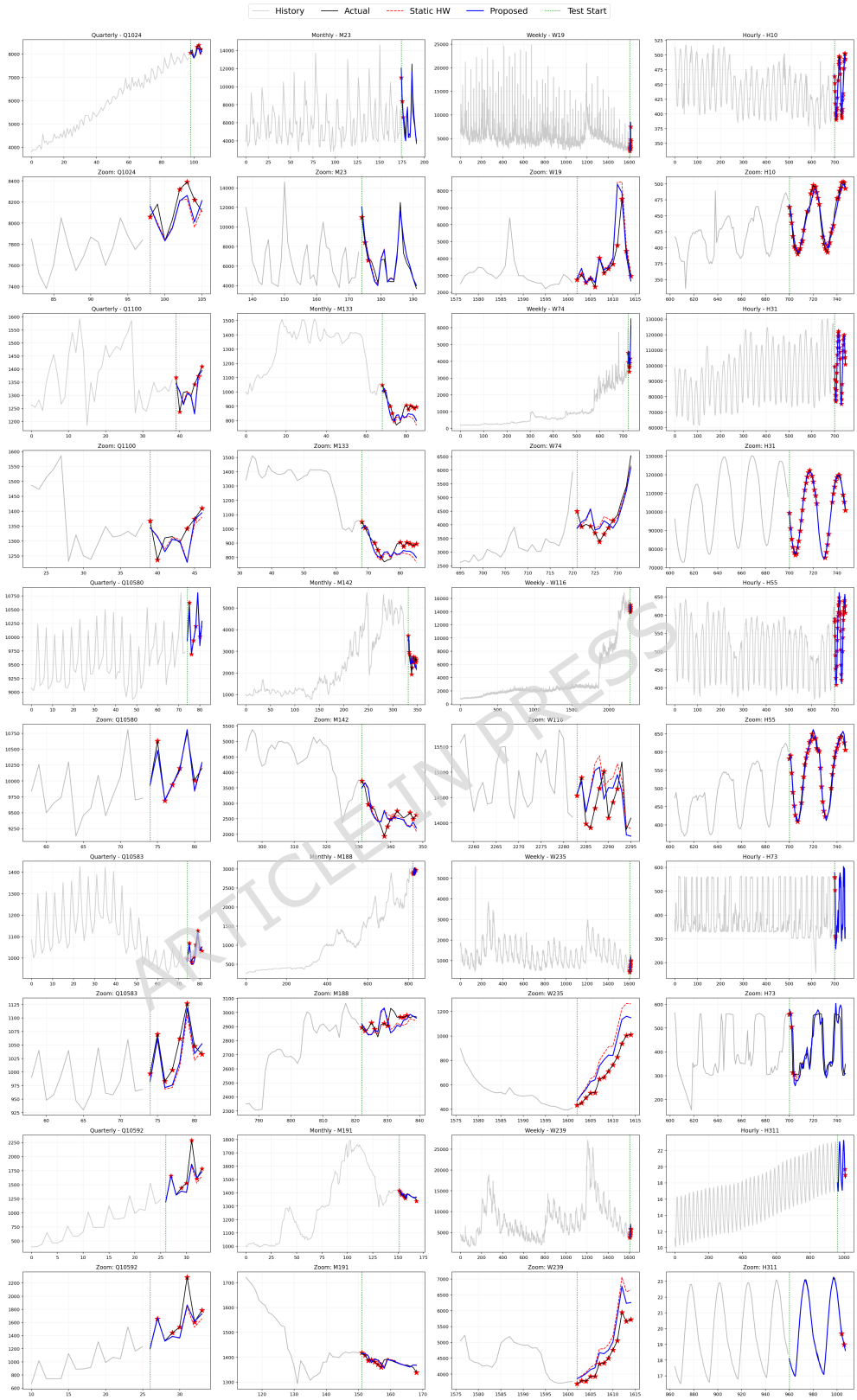


Fig. 8 Representative one-step-ahead forecasting results for Quarterly, Monthly, Weekly, and Hourly series, comparing the proposed SETS-HW method (blue) against the static HW model (red dashed). Red stars indicate points where the SETS mechanism triggered an adaptive bias correction

ARTICLE IN PRESS

6 Conclusion

This study introduced a novel WES method enhanced with three key modifications: back-casting for initialization, GWO for parameter tuning, and SETS for dynamic adaptation. The proposed method was evaluated on both additive (average temperature data from Germany) and multiplicative (time series from the R 'tsdl' package) time series data, as well as a comprehensive benchmark of 40 series from the M4 Forecasting Competition. Performance, measured by RMSE, MAE, MAPE, and SMAPE, was compared against standard WES implementations in SPSS, Python, and R, along with automated ARIMA and ETS frameworks.

Results demonstrated that the proposed method consistently outperformed the benchmark software packages and statistical models. Ablation studies highlighted the contribution of each modification, with SETS playing a crucial role in achieving the most significant performance gains. Even without SETS, the proposed method generally outperformed existing implementations, showcasing the benefits of back-casting and GWO. Notably, the integration of SETS enabled the model to adapt effectively to forecast errors, leading to superior accuracy particularly with shorter training lengths and high-frequency data. Statistical analyses confirmed the significance of these findings, with the proposed method achieving a superior mean rank across various datasets and temporal frequencies.

However, several practical considerations and limitations must be acknowledged. First, GWO relies on a stochastic search process, which prevents perfectly consistent outcomes across independent runs, unlike deterministic tools. Second, the framework's sensitivity to the training length in the monitoring phase and the selection of the SETS threshold remains a challenge; improper calibration risks either underfitting or delayed error detection. Furthermore, while the framework proved computationally faster than automated ARIMA models, this study did not explicitly test robustness under artificially injected structural breaks or perform a deep audit of energy consumption and algorithmic efficiency. From an ethical perspective, applying this model to socio-economic datasets requires caution regarding potential bias implications, which may be more critical in sensitive domains such as healthcare or financial risk assessment.

Future work will focus on addressing these gaps by incorporating nested validation for automated hyperparameter selection and systematically testing the model's resilience against synthetic structural shifts. Additionally, exploring the scalability of the proposed method for high-dimensional "Big Data" streaming applications and analyzing the algorithmic efficiency in resource-constrained environments will broaden the practical utility of the enhanced WES framework.

ARTICLE IN PRESS

Declarations

Funding. No funding was received for this study.

Conflict of interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ARTICLE IN PRESS

Data availability

The datasets analyzed during the current study are derived from the following public repositories:

1. The additive time series data (regional air temperature mean) are provided by the German Weather Service (DWD) and were accessed on February 20, 2025, through the Climate Data Center (https://opendata.dwd.de/climate_environment/CDC/regional_averages_DE/monthly/air_temperature_mean/). The data are licensed under the GeoNutzV ordinance.
2. The multiplicative time series data were obtained from the Time Series Data Library (TSDL), accessed on February 20, 2025, via the 'tsdl' R package (version 0.1.0) available at https://github.com/FinYang/tsdl.
3. The M4 Competition dataset was accessed on March 6, 2026, via the 'datasetsforecast' Python library (version 0.0.8, https://pypi.org/project/datasetsforecast), which provides an interface to the original M4 dataset (DOI: https://doi.org/10.1016/j.ijforecast.2018.06.001).

All datasets used in this study are publicly available.

Code availability

The analysis code and the specific version of the framework used to produce the results reported in this study are permanently archived in Zenodo at https://doi.org/10.5281/zenodo.19390159 (DOI: 10.5281/zenodo.19390159). The source code is also available in a public GitHub repository at https://github.com/erdincK-61/Enhanced_WES_Adaptive_GWO. All computational materials are shared under the MIT License to ensure full reproducibility.

References

Alam, T., AlArjani, A.: A comparative study of co2 emission forecasting in the gulf countries using autoregressive integrated moving average, artificial neural network, and holt-winters exponential smoothing models. Advances in Meteorology (1), 8322590 (2021) https://doi.org/10.1155/2021/8322590

Al-Aboody, N.A., Al-Raweshidy, H.S.: Grey wolf optimization-based energy-efficient routing protocol for heterogeneous wireless sensor networks. In: 2016 4th International Symposium on Computational and Business Intelligence (ISCBI), pp. 101–107 (2016). https://doi.org/10.1109/ISCBI.2016.7743266

Alhussan, A., Abdelhamid, A., El-kenawy, E.-S., Ibrahim, A., Eid, M., Khafaga, D., Ahmed, A.: A binary waterwheel plant optimization algorithm for feature selection. IEEE Access **11** (2023) https://doi.org/10.1109/ACCESS.2023.3312022

Bilal, A., Imran, A., Baig, T.I., Liu, X., Abouel Nasr, E., Long, H.: Breast cancer diagnosis using support vector machine optimized by improved quantum inspired

ARTICLE IN PRESS

grey wolf optimization. Sci. Rep. **14**(1), 10714 (2024) https://doi.org/10.1038/s41598-024-61322-w

Bianchesi, N.M., Matta, C., Streitenberger, S., Romão, E., Balestrassi, P., Costa, A.: A nonlinear time-series prediction methodology based on neural networks and tracking signals. Production **32** (2022) https://doi.org/10.1590/0103-6513.20220064

Brown, R.G.: Statistical Forecasting for Inventory Control. McGraw-Hill, New York (1959)

Cañedo, M.C., Lopes, T.I.B., Rossato, L., Nunes, I.B., Faccin, I.D., Salomé, T.M., Simionatto, S.: Impact of covid-19 pandemic in the brazilian maternal mortality ratio: A comparative analysis of neural networks autoregression, holt-winters exponential smoothing, and autoregressive integrated moving average models. Plos one **19**(1), 0296064 (2024) https://doi.org/10.1371/journal.pone.0296064

Deng, Y., Jaraiedi, M., Iskander, W.H.: Tracking signal test to monitor an intelligent time series forecasting model. In: Intelligent Manufacturing, vol. 5263, pp. 149–160 (2004). https://doi.org/10.1117/12.517225

DWD: Deutscher Wetterdienst - German Meteorological Service. [Accessed: March 2024] (2024). https://opendata.dwd.de/climate_environment/CDC/regional_averages_DE/monthly/air_temperature_mean

El-Kenawy, E.-S., Ibrahim, A., Mirjalili, S., Zhang, Y., Elnazer, S., Mounir, R.: Optimized ensemble algorithm for predicting metamaterial antenna parameters. Computers, Materials and Continua **71**, 4989–5003 (2022) https://doi.org/10.32604/cmc.2022.023884

El-Kenawy, E.-S., Mirjalili, S., Khodadadi, N., Abdelhamid, A., Eid, M., Said, E.M., Ibrahim, A.: Feature selection in wind speed forecasting systems based on meta-heuristic optimization. PLOS ONE **18**, 0278491 (2023) https://doi.org/10.1371/journal.pone.0278491

Eswaramoorthy, S., Sivakumaran, N., Sekaran, S.: Grey wolf optimization based parameter selection for support vector machines. COMPEL-The international journal for computation and mathematics in electrical and electronic engineering **35**(5), 1513–1523 (2016) https://doi.org/10.1108/COMPEL-09-2015-0337

Gardner Jr, E.S.: Cusum vs smoothed-error forecast monitoring schemes: Some simulation results. Journal of the Operational Research Society **36**(1), 43–47 (1985) https://doi.org/10.1057/jors.1985.6

Gardner Jr, E.S.: Exponential smoothing: The state of the art—part ii. International journal of forecasting **22**(4), 637–666 (2006) https://doi.org/10.1016/j.ijforecast.2006.03.005

ARTICLE IN PRESS

Gardner Jr, E.S., McKenzie, E.: Forecasting trends in time series. Management science **31**(10), 1237–1246 (1985)

Hansun, S.: New estimation rules for unknown parameters on holt-winters multiplicative method. Journal of Mathematical and Fundamental Sciences **49**(2), 127–135 (2017) https://doi.org/10.5614/j.math.fund.sci.2017.49.2.3

Hyndman, R.J., Koehler, A.B., Snyder, R.D., Grose, S.: A state space framework for automatic forecasting using exponential smoothing methods. International Journal of forecasting **18**(3), 439–454 (2002) https://doi.org/10.1016/S0169-2070(01)00110-8

IBM SPSS: IBM SPSS Statistics Algorithms. IBM Corporation. [Accessed: March 2024] (2016). https://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/en/client/Manuals/IBM_SPSS_Statistics_Algorithms.pdf

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 **193**, 116779 (2020) https://doi.org/10.1016/j.energy.2019.116779

Kinast, K.B., Fokoué, E.: An in-depth look at rising temperatures: Forecasting with advanced time series models in major us regions. Forecasting **6**(3), 815–838 (2024) https://doi.org/10.3390/forecast6030041

Kristianto, R.P.: Modeling of time series data prediction using fruit fly optimization algorithm and triple exponential smoothing. In: 2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), pp. 407–412 (2019). https://doi.org/10.1109/ICITISEE48480.2019.9003895

Liu, Y., As' arry, A., Ahmed, H., Hairuddin, A.A., Hassan, M.K., Zakaria, M.Z., Yang, S.: Online optimal tuning of fuzzy pid controller using grey wolf optimizer for quarter car semi-active suspension system. Advances in Mechanical Engineering **16**(2), 16878132231219620 (2024) https://doi.org/10.1177/16878132231219620

Li, Q., Chen, H., Huang, H., Zhao, X., Cai, Z., Tong, C., Liu, W., Tian, X.: An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis. Computational and mathematical methods in medicine **2017**(1), 9512741 (2017) https://doi.org/10.1155/2017/9512741

Li, Y., Lange, M., Stocks, C.: Monitoring forecasting systems–revisit trigg's tracking signal. In: The 32nd Annual International Symposium on Forecasting, pp. 24–27 (2012)

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R.: The accuracy of extrapolation (time series) methods: Results of a forecasting competition. Journal of forecasting **1**(2), 111–153

ARTICLE IN PRESS

(1982) https://doi.org/10.1002/for.3980010202

McClain, J.O.: Dominant tracking signals. International Journal of Forecasting **4**(4), 563–572 (1988) https://doi.org/10.1016/0169-2070(88)90133-1

Mirjalili, S.: How effective is the grey wolf optimizer in training multi-layer perceptrons. Applied intelligence **43**, 150–161 (2015) https://doi.org/10.1007/s10489-014-0645-7

Montgomery, D.C., Johnson, L.A., Gardiner, J.S.: Forecasting and Time Series Analysis. McGraw-Hill, New York (1990)

Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in engineering software **69**, 46–61 (2014) https://doi.org/10.1016/j.advengsoft.2013.12.007

Mauricio, C.C., Ostia, C.F.: Cuckoo search algorithm optimization of holt-winter method for distribution transformer load forecasting. In: 2023 9th International Conference on Control, Automation and Robotics (ICCAR), pp. 36–42 (2023). https://doi.org/10.1109/ICCAR57134.2023.10151700

Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The m4 competition: 100,000 time series and 61 forecasting methods. International Journal of Forecasting **36** (2019) https://doi.org/10.1016/j.ijforecast.2019.04.014

Medjahed, S.A., Saadi, T.A., Benyettou, A., Ouali, M.: Gray wolf optimizer for hyperspectral band selection. Applied Soft Computing **40**, 178–186 (2016) https://doi.org/10.1016/j.asoc.2015.09.045

Nembhard, H.B., Chen, S.: Cuscore control charts for generalized feedback-control systems. Quality and Reliability Engineering International **23**(4), 483–502 (2007) https://doi.org/10.1002/qre.831

Nembhard, H.B., Kao, M.S.: A forecast-based monitoring methodology for process transitions. Quality and Reliability Engineering International **17**(4), 307–321 (2001) https://doi.org/10.1002/qre.403

Rdocumentation: HoltWinters: Holt-Winters Filtering. [Accessed: March 2024] (2024). https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/HoltWinters

Shahin, A.A.: Using multiple seasonal holt-winters exponential smoothing to predict cloud resource provisioning. International Journal of Advanced Computer Science and Applications **7**(11), 91–96 (2016) https://doi.org/10.14569/IJACSA.2016.071113

Seabold, S., Perktold, J.: Statsmodels: Econometric and Statistical Modeling with Python. In: Walt, Millman (eds.) Proceedings of the 9th Python in Science

ARTICLE IN PRESS

Conference, pp. 92–96 (2010). https://doi.org/10.25080/Majora-92bf1922-011

Sarma, R., Singh, S.: A comparative study of data-driven models for groundwater level forecasting. Water Resources Management **36**(8), 2741–2756 (2022) https://doi.org/10.1007/s11269-022-03173-6

Song, X., Tang, L., Zhao, S., Zhang, X., Li, L., Huang, J., Cai, W.: Grey wolf optimizer for parameter estimation in surface waves. Soil Dynamics and Earthquake Engineering **75**, 147–157 (2015) https://doi.org/10.1016/j.soildyn.2015.04.004

Segura, J.V., Vercher, E.: A spreadsheet modeling approach to the holt–winters optimal forecasting. European Journal of Operational Research **131**(2), 375–388 (2001) https://doi.org/10.1016/S0377-2217(00)00062-X

Tak, N.: A novel arma type possibilistic fuzzy forecasting functions based on grey-wolf optimizer (arma-pffs). Computational Economics **59**(4), 1539–1556 (2022) https://doi.org/10.1007/s10614-021-10132-7

Takieldeen, A.E., El-kenawy, E.-S.M., Hadwan, M., Zaki, R.M.: Dipper throated optimization algorithm for unconstrained function and feature selection. Computers, Materials and Continua **72**(1), 1465–1481 (2022) https://doi.org/10.32604/cmc.2022.026026

Trull, O., García-Díaz, J.C., Troncoso, A.: Initialization methods for multiple seasonal holt-winters forecasting models. Mathematics **8**(2) (2020) https://doi.org/10.3390/math8020268

Trigg, D.: Monitoring a forecasting system. OR **15**(3), 271–274 (1964) https://doi.org/10.2307/3007215

Winters, P.R.: Forecasting sales by exponentially weighted moving averages. Management science **6**(3), 324–342 (1960) https://doi.org/10.1287/mnsc.6.3.324

Wang, J.S., Li, S.X.: An improved grey wolf optimizer based on differential evolution and elimination mechanism. Scientific Reports **9**, 7181 (2019) https://doi.org/10.1038/s41598-019-43546-3

Wongoutong, C.: Improvement of the holt-winters multiplicative method with a new initial value settings method. Thailand Statistician **19**(2), 280–293 (2021)

Wisner, J., Tan, K.-C., Leong, G.K.: Principles of Supply Chain Management: A Balanced Approach. South-Western, Mason, OH (2009)

Yaffee, R.A., McGee, M.: Introduction to Time Series Analysis and Forecasting: with Applications of SAS and SPSS. Academic Press, San Diego, CA (2000)

Yu, L., Wang, S., Lai, K.K.: Adaptive smoothing neural networks in foreign exchange

ARTICLE IN PRESS

rate forecasting. In: International Conference on Computational Science, pp. 523–530 (2005). https://doi.org/10.1007/11428862_72

ARTICLE IN PRESS