Journal of Mathematical Modeling
Vol. 11, No. 4, 2023, pp. 783–803. Research Article

JMM

A comparative study of singular spectrum analysis, neural network, ARIMA and exponential smoothing for monthly rainfall forecasting

Mohammad Kazemi*

Department of Statistics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran
Email(s): m.kazemi@guilan.ac.ir

Abstract. This paper investigates the accuracy of several forecasting methods for monthly rainfall forecasting. First, we study the feasibility of using the Singular Spectrum Analysis (SSA) to perform rainfall forecasts. When the time series data has the outliers, SSA might results in misleading conclusions, and thus robust methodologies should be used. Therefore, we consider the use of two robust SSA algorithms for model fit and model forecasting. The results of these forecasting approaches are compared with other commonly used time series forecasting techniques including Neural Network Autoregression (NNAR), Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing (ETS) and TBATS. The performance of these conjunction methods is compared in terms of accuracy for model fit and model forecast, using the monthly rainfall data from four rain gauge stations in Guilan province of Iran as the case study.

Keywords: Contaminated data, forecasting, singular value decomposition, time series.
AMS Subject Classification 2010: 60G35, 62M10, 62M20.

1 Introduction

Rainfall, as an essential process in the hydrological cycle, is one of the most studied components of hydrological and climate science, as it directly or indirectly affects our society. Accurate rainfall forecasting is vital in daily life, risk assessment, natural disaster prevention, and water resource planning and management [33]. However, it is a difficult task due to the dynamic complexity and nonstationary nature of measured hydrological data [1].

Several techniques for forecasting time series have been developed on a global scale. Stochastic models and Artificial Intelligence (AI) are the most widely used time series modeling approaches for

*Corresponding author
Received: 30 August 2023 / Revised: 6 October 2023 / Accepted: 13 October 2023
DOI: 10.22124/jmm.2023.25412.2262

hydrological forecasting. In stochastic modeling, forecasts are deciphered based on the statistical characteristics of the past data [2]. Autoregressive Integrated Moving Average (ARIMA) model, which is the most widely used stochastic model for forecasting time series, has great flexibility. In the case of stochastic models, for yielding reliable results, the data has to be stationary [24]. A homogeneous non-stationary time series can be reduced to a stationary time series, by taking a proper degree of differencing. However, in the ARIMA model, the differencing can reduce only small-scale nonstationary process to a stationary process [40]. AI-based time series models have gained popularity over the last few decades. These models are based on the input-output relationships. The most widely used AI-based models include Artificial Neural Network (ANN), Genetic Programming (GP), and Model Tree (MT). Among them, ANN has gained widespread acceptance from researchers in various fields. The ANN models are highly flexible so that, any combination of different algorithms can be developed according to the complexity of the data. However, the major drawback of ANN is that it is a grey box model, and outliers present in the data can critically affect the reliability of the model. It is also reported by many researchers that, for climatic data to yield reliable results by ANN, the data need to be preprocessed [32].

Singular Spectrum Analysis (SSA) is a powerful non-parametric technique for time series analysis and forecasting, which incorporates elements of classical time series analysis, multivariate statistics, and matrix algebra. Its main aim is to decompose the original time series into a set of components that can be interpreted as trend components, seasonal components, and noise components [4, 7, 8]. The SSA has proven both wide usefulness and applicability across many applications [10, 27, 41, 46], being that its scope of application ranges from parameter estimation to time series filtering, synchronization analysis, and forecasting.

The SSA methodology for model fit can be summarized in four steps: (i) embedding, which maps the original univariate time series into a trajectory matrix; (ii) singular value decomposition (SVD), which helps decomposing the trajectory matrix into the sum of rank-one matrices; (iii) eigentriple grouping, which helps deciding which of the components are associated to the signal and which are associated to the noise; and (iv) diagonal averaging, which maps the rank-one matrices, associated to the signal, back to time series that can be interpreted as trend, seasonal, or other meaningful components.

The SSA results and interpretation, similarly to many other classical time series methods, can be sensitive to data contamination with outliers [34, 35]. In those cases, even a small percentage of outliers can make a big difference in the results for model fit and forecast. Very few attempts have been made in order to access the effect of the presence of outliers in the data while conducting SSA. The authors in [11, 36] presented some preliminary results on the effect of outliers in SSA, and in [38] the first attempt is made to robustify SSA by considering an SVD based on a robust L_1 norm instead of the L_2 norm used in the classical algorithm, which they used for model fit. The authors in [37] proposed a robust algorithm for SSA that considers the SVD based on the Huber function. Also, in [20] four robust alternatives to the SSA are proposed using the robust regularized SVD and the robust principal component analysis algorithms.

In this paper, we focus on the performance of various SSA forecasting algorithms when applied to the time series of monthly rainfall. In order to evaluate the potential of SSA for this aim, the performance of SSA and its robust alternatives are compared with other commonly used time series forecasting techniques including Neural Network Autoregression (NNAR), ARIMA, Exponential Smoothing (ETS), and TBATS that stands for Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend and Seasonal components. These comparisons are done by considering the monthly rainfall data from four rain gauge stations in Guilan province of Iran. The selection of this location is driven by the increasing

of immigrants from various provinces of Iran to Guilan. This demographic shift raises concerns over potential water scarcity challenges that may arise in the future. Hence, accurate forecasting of rainfall levels in Guilan province assumes paramount importance for effective water resource management.

The rest of this paper is organized as follows. In Section 2, a brief description of the SSA algorithm is presented. Section 3 focuses on the robust SSA methodology. In Section 4, the general scheme of other time series forecasting techniques utilized in this study are briefly discussed. Section 5 presents the results, wherein SSA, and robust SSA algorithms are compared with other forecasting methods in terms of model fit and forecast, using the four monthly rainfall dataset. Finally, the concluding remarks are drawn in Section 6.

2 Singular Spectrum Analysis (SSA)

The SSA has been recognized as a powerful technique for analyzing nonlinear and nonstationary time series. The primary aim of SSA is to decompose the original series into the sum of a small number of independent and interpretable components such as a trend components (which may not exist), oscillatory components and noise [7]. The SSA technique has various modifications and extensions, some of them are discussed in [1, 8]. The most fundamental version of SSA is called Basic SSA, in which we briefly explain the theory underlying it. Also, one of the SSA forecasting methods namely *recurrent forecasting* is briefly reviewed.

2.1 Basic SSA algorithm

The SSA technique consists of two complementary stages: decomposition and reconstruction. Each of these stages includes two separate steps. At the decomposition stage, a time series is decomposed into several interpretable components such as trend, seasonal and cyclical components, enabling us to signal extraction and noise reduction. At the reconstruction stage, interpretable components are reconstructed, which can be used to forecast new data points.

2.1.1 First stage: decomposition

The decomposition stage is performed in two sequential steps. First, the time series is converted into a high-dimensional matrix called *trajectory matrix*. Then the trajectory matrix is decomposed into the sum of rank-one matrices using SVD.

First step (embedding): Let $Y_N = [y_1, \ldots, y_N]$ be a time series of length N. For a given window length L, the result of this step is a $L \times K$ matrix $\mathbf{X} = [Y_1, \ldots, Y_K]$, where $K = N - L + 1$ and $Y_i = (y_i, \ldots, y_{i+L-1})^T, 1 \leq i \leq K$. This matrix is often called *trajectory matrix*. It is a Hankel matrix, which means that all the elements along the diagonal $i + j = constant$ are equal.

Second step (SVD): Let $\mathbf{S} = \mathbf{X}\mathbf{X}^T$, and denote by $\lambda_1, \ldots, \lambda_d$ the positive eigenvalues of $\mathbf{S}$ in the decreasing order of magnitude ($\lambda_1 \geq \cdots \geq \lambda_d > 0$) and $U_1, \ldots, U_d$ be the orthonormal eigenvectors of the corresponding eigenvalues of $\mathbf{S}$. In this step, the trajectory matrix $\mathbf{X}$ will be decomposed using SVD as:

$$\mathbf{X} = \mathbf{X}_1 + \cdots + \mathbf{X}_d, \tag{1}$$

where $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$ and $V_i = \mathbf{X}^T U_i / \sqrt{\lambda_i}$ $(i = 1, \ldots, d)$. The collection $(\sqrt{\lambda_i}, U_i, V_i)$ is called i-th eigentriple of the SVD (1).

2.1.2 Second stage: reconstruction

In the second stage, a diagonal averaging procedure is conducted in the matrices associated to the signal resulting into the sum of time series components that can then be interpreted as trend or oscillatory components.

First step (eigentriple grouping): In this step, two main groups are created, one with components associated to the signal and another with components associated to the noise. Formally, let $I = 1, \ldots, r$ and $I^c = r+1, \ldots, d$. Here, the first r leading eigentriples associated to the signal are chosen, while excluding the remaining $d - r$ eigentriples associated to the noise. Therefore, the trajectory matrix can be written as:

$$\mathbf{X} = \sum_{i \in I} \mathbf{X}_i + \varepsilon = \sum_{i \in I} \sqrt{\lambda_i} U_i V_i^T + \varepsilon, \tag{2}$$

where ε is the noise term, being the noise-free approximation of the trajectory matrix written as $\mathbf{X}_I = \sum_{i \in I} \mathbf{X}_i$.

Second step (diagonal averaging): In this step, using anti-diagonal averaging on the matrices included in $\mathbf{X}_I$, the noise-free time series is reconstructed. First, the approximate trajectory matrix $\mathbf{X}_I$ is transformed into a Hankel matrix. Let $A_s = \{(l,k) : l + k = s, 1 \leq l \leq L, 1 \leq k \leq K\}$ and $\#(A_s)$ be the number of elements in A_s. The element $\widetilde{x}_{ij}$ of the new Hankel matrix $\widetilde{\mathbf{X}}_I$ is given by

$$\widetilde{x}_{ij} = \sum_{(l,k) \in A_s} \frac{x_{lk}}{\#(A_s)}. \tag{3}$$

Next, the Hankel matrix $\widetilde{\mathbf{X}}_I$ is transformed into a new series of dimension N, and the original time series Y_N can be approximated by:

$$\widetilde{y}_i = \begin{cases} \widetilde{x}_{i1} & \text{for } i = 1, \ldots, L, \\ \widetilde{x}_{Lj} & \text{for } i = L+1, \ldots, N, \end{cases} \tag{4}$$

where $j = i - L + 1$. The reconstructed noise-fee time series can then be used for out-of-sample forecasting.

2.2 SSA forecasting

In SSA, there are two main forecasting methods: recurrent SSA and vector SSA. Here, we consider the first one that is explained as follows.

Let $U_j^{\triangledown}$ denotes the vector of the first $L-1$ components of U_j, the j-th eigenvector of $\mathbf{X}\mathbf{X}'$, π_j denotes the last component of U_j, $j = 1, \ldots, r$, and r denotes the number of eigenvalues used for reconstruction. We can define the coefficient vector $\widehat{a}$ as

$$\widehat{a} = (\widehat{a}_{L-1}, \ldots, \widehat{a}_1)' = \frac{1}{1 - \nu^2} \sum_{j=1}^{r} \pi_j U_j^{\triangledown}, \tag{5}$$

where $\nu^2 = \sum_{j=1}^{r} \pi_j^2$. Considering the above notation, the h steps ahead out-of-sample recurrent SSA forecasts $\widehat{y}_{N+1}, \ldots, \widehat{y}_{N+h}$ can be obtained as

$$\widehat{y}_t = \begin{cases} \widetilde{y}_t & \text{for } t = 1, \ldots, N, \\ \sum_{j=1}^{L-1} \widehat{a}_j \widehat{y}_{t-j} & \text{for } t = N+1, \ldots, N+h, \end{cases} \tag{6}$$

where $\widetilde{y}_1, \ldots, \widetilde{y}_N$, are the fitted values for the reconstructed time series as obtained from equation (4).

2.3 SSA parameters selection

The SSA calibration depends upon two important parameters: the window length L, and the number of eigentriples used for reconstruction r. The improper choice of L would imply an inferior decomposition and incomplete reconstruction and misleading results in forecasting [26, 44]. Setting L parameter too large could lead to the noise mixing up with the signal, and choosing L too small opens up the risk of losing some parts of the signal to the noise. In [21], the authors recommended that L could be $\frac{N}{2}$ or $\frac{N}{4}$ to achieve optimal signal-noise separation and get better SSA forecasts. However, there are no specific rules of selecting L parameter as it is depends on the structure of time series and the purpose of data analysis.

Additionally, the large number of eigentriples r increase the noise in the reconstructed series. Also, it might miss some parts of the signal when we consider r smaller than what is supposed to be [12]. Among several ways to determine r described in the literature, the easiest way is done by checking breaks in the eigenvalues spectra. As a rule of thumb, a pure noise series produces a slowly decreasing sequences of singular values. Another useful insight is provided by considering separability between signal and noise components, which is a fundamental concept in studying SSA properties, by using w-correlations [7]. We shall say that two series $Y^{(1)}$ and $Y^{(2)}$ are approximately separable if all correlations between the rows and the columns of the corresponding trajectory matrices obtained from series $Y^{(1)}$ and $Y^{(2)}$ are close to zero. In [7] they considered other characteristics of the quality of separability; namely, the weighted correlation or w-correlation, which is a natural measure of deviation of two series $Y_T^{(1)}$ and $Y_T^{(2)}$ from w-orthogonality:

$$\rho_{12}^{(w)} = \frac{\left(Y_T^{(1)}, Y_T^{(2)}\right)_w}{\|Y_T^{(1)}\|_w \|Y_T^{(1)}\|_w}, \tag{7}$$

where $\|Y_T^{(i)}\|_w = \sqrt{\left(Y_T^{(i)}, Y_T^{(i)}\right)_w}$, $i = 1, 2$, and $\left(Y_T^{(1)}, Y_T^{(2)}\right)_w = \sum_{t=1}^{T} w_t y_t^{(1)} y_t^{(2)}$ with $w_t = \min\{t, L, T-t+1\}$.

According to this measure, two series are separable if the absolute value of their w-correlation is small. Therefore, we determine r in such a way that the reconstructed series and residual have a small w-correlation. Another way to determine r is by examining the forecasting accuracy, i.e. r is determined in such a way that the minimum error in forecasting will be obtained. For other proposals one may see [25, 39].

3 Robust SSA

Despite knowing that SSA has shown to be superior to traditional model-based methods in many applications, the SVD (second step of the SSA algorithm) is highly sensitive to data contamination with outliers.

A first attempt to robustify the SSA by considering an SVD based on a robust L_1 norm instead of the L_2 norm used in the classical algorithm, is proposed by [38]. In [37] the authors proposed another robust algorithm for SSA considering the SVD based on the Huber function and also suggested an algorithm for robust SSA model forecasting. In this section, we briefly review those robust SSA algorithms.

3.1 Robust SSA based on the L_1 norm

The robust SSA algorithm proposed by [38] replaces the classical SVD based on the least squares L_2 norm, by the robust SVD algorithm based on the L_1 norm. This robust SVD is performed iteratively, starting with an initial estimate of the first left singular vector U_1 and leading to an outlier-resistant approach that also allows for missing data. The algorithm for the L_1 alternating procedure can be found in [13] and a nice flowchart of the algorithm is presented by [22]. The robust SVD based on the L_1 norm is implemented using the function `robustSVD` from the `R` package `pcaMethods`.

3.2 Robust SSA based on the Huber function

Another robust alternative to the SSA algorithm is obtained by the robust SVD based on the Huber function [14]. It is well known that SVD can be viewed as finding a sequence of rank-one matrix approximations of a data matrix [5]. This idea is adapted to define a method for obtaining a sequence of robust rank-one matrix approximations [45]. Our discussion focuses on obtaining the first pair of components. Subsequent pairs of components can be obtained by applying the method sequentially on the residuals from lower rank approximations. In SVD, the first pair of singular vectors of a data matrix $\mathbf{X} = (x_{ij})_{m\times n}$ can be obtained by solving a least squares problem as:

$$(\hat{u},\hat{v}) = \arg\min_{u,v}(||\mathbf{X} - uv'||_F^2), \tag{8}$$

where u and v are $m\times 1$ and $n\times 1$ vectors, respectively, and $||.||_F$ is the Frobenius norm of a matrix. To achieve robustness, we replace the quadratic loss function in (8) with the Huber function [14] as follows:

$$L_\delta(x) = \begin{cases} \frac{1}{2}x^2, & \text{if } |x| \le \delta, \\ \delta\left(|x| - \frac{1}{2}\delta\right), & \text{if } |x| > \delta, \end{cases} \tag{9}$$

where δ is a parameter that controls the robustness level, and a smaller value of δ usually leads to more robust estimation. Thus, the first pair of singular vectors of the data matrix $\mathbf{X}$ can be defined as follows:

$$(\hat{u},\hat{v}) = \arg\min_{u,v} L_\delta\left(\frac{\mathbf{X} - uv'}{\sigma}\right), \tag{10}$$

where σ is the scale parameter measuring the variability in the approximation errors. In practice, σ can be estimated from the data using residuals from a preliminary rank-one approximation of $\mathbf{X}$, and we refer to [45] for more detatils.

With a slight abuse of notation, we also use $L_\delta(.)$ to denote the summation over elementwise applications when the scalar function $L_\delta(.)$ is applied to a matrix. A general loss function for rank-one approximation of the matrix $\mathbf{X}$ can be written as:

$$L_\delta\left(\frac{\mathbf{X} - uv'}{\sigma}\right) = \sum_{i=1}^{m}\sum_{j=1}^{n} L_\delta\left(\frac{x_{ij} - u_i v_j}{\sigma}\right). \tag{11}$$

To solve the optimization problem (10) an iterative reweighted least squares (IRLS) algorithm is used [45].

The robust SVD based on the Huber function is a special case of robust regularized SVD, and can be obtained with the function `RobRSVD` from the `R` package `RobRSVD`. In this R implementation, the authors consider $\delta = 1.345$, the value commonly used in robust regression that produces 95% efficiency for normal errors. More details about this robust SVD can be found in [45].

4 Other forecasting methods

In this section, the other commonly used time series forecasting methods applied in this investigation are briefly explained.

4.1 Neural Network Autoregression (NNAR)

There has been a growing interest in using neural networks for modeling and forecasting time series data. A neural network can be considered as a network of neurons which are arranged in layers. In this structure, the first layer consists of predictors or inputs, while the last layer comprises forecasts or outputs. Additionally, there may exist hidden layers that contain some neurons. Without any hidden layers, the network is equivalent to a linear regression. However, the introduction of an activation function adds non-linearity to the neural network.

This study focuses on a multilayer feed-forward network known as NNAR. To provide a complete overview of this model, we will primarily refer to [18] and [43]. The NNAR model utilizes lagged values of a time series as inputs to a neural network. The notation $NNAR(p,k)$ is used in [18] to indicate feed-forward networks with one hidden layer, p lagged inputs and k nodes in the hidden layer. Additionally, a seasonal NNAR model is denoted by $NNAR(p,P,k)_m$ where p represents the number of lagged inputs, P denotes the number of seasonal lags, and m represents the number of periods. The inputs for this NNAR model consist of lagged values of the time series, inclusive of both lagged and seasonal lags. To be more specific, the inputs can be represented as $y_{t-1}, y_{t-2}, \ldots, y_{t-p}, y_{t-m}, y_{t-2m}, \ldots, y_{t-Pm}$. In the NNAR model, the inputs into each hidden layer neuron are combined linearly to give weight and produce output from artificial neural networks and the activation function as the binary sigmoid, which is a nonlinear function. Specifically, the j-th hidden layer neuron is defines as follows:

$$Z_j = \alpha_j + \sum_{i=1}^{N} \omega_{i,j} y_i,$$

where N represents the number of input layer neurons, α_j represents the intercept of the j-th hidden neuron, $\omega_{i,j}$ denotes the weights assigned to the connection between the input and the hidden layer, y_i's are the covariates or neurons of the input layer, and the activation function is given by

$$g(z) = \frac{1}{1+e^{-z}}.$$

We consider the algorithm proposed by Hyndman [15] that defines the number of nodes in the hidden layer (k) as an average of the number of inputs and the number of outputs, that is, $(p+P+1)/2$. The `nnetar` function in the `forecast` package of `R` software fits an $NNAR(p,P,k)_m$ model to time series data. In this function, the values of p and P are selected automatically if they are not specified.

4.2 Autoregressive Integrated Moving Average model (ARIMA)

The ARIMA model is among the most widely used techniques for time series analysis and forecasting. The non-seasonal ARIMA model depends on three parameters: p is the number of lagged observations in the model, i.e., the autoregressive (AR) order; d is the number of times that the original observations are differenced, i.e., the integrated (I) degree; and q is the size of the moving average window, i.e., the order of the moving average (MA) [3]. This parametric model can then be written as $ARIMA(p,d,q)$, with p, d, and q non-negative integers. Given a time series $Y_N = [y_1,\ldots,y_N]$, the $ARIMA(p,d,q)$ model can be written as:

$$\phi(B)(1-B)^d y_t = c + \theta(B)\varepsilon_t, \tag{12}$$

where y_t is the observation at the time point t; B is the time lag operator, or backward shift, which is a linear operator denoted by B^k such that $B^k y_t = y_{t-k}$; $\phi(B) = 1-\phi_1 B^1 - \cdots - \phi_p B^p$; $\theta(B) = 1+\theta_1 B^1 + \cdots + \theta_q B^q$; $c = \mu(1-\phi_1-\cdots-\phi_p)$; μ is the mean of $(1-B)^d y_t$; and ε_t is an error term, usually white noise with mean zero. The seasonal ARIMA model incorporates both non-seasonal and seasonal factors in a multiplicative model. The seasonal $ARIMA(p,d,q)(P,D,Q)_m$ model is written as

$$\Phi(B^m)\phi(B)(1-B^m)^D(1-B)^d y_t = c + \Theta(B^m)\theta(B)\varepsilon_t, \tag{13}$$

where $\Phi(B^m) = 1-\Phi_1 B^m - \cdots - \phi_P B^{Pm}$; $\Theta(B^m) = 1+\theta_1 B^m + \cdots + \theta_Q B^{Qm}$; p is the non-seasonal AR order, d is the non-seasonal differencing, q is the non-seasonal MA order, P is the seasonal AR order, D is the seasonal differencing, Q is the seasonal MA order, and m is time span of repeating seasonal pattern.

Selecting an appropriate model order, that is the values p,d,q,P,D and Q, is a major task in ARIMA modeling. In this paper, we use the `auto.arima` function from the `forecast` package of `R` software to find the best ARIMA model automatically and estimate its parameters. For more information on how this function works and examples of applications, see [17].

4.3 Exponential smoothing

Exponential smoothing methods are among the most widely used forecasting procedures in practice due to their simplicity and effectiveness. These were originally classified by Pegels [31] and later modified by Hyndman et al. [19], and extended by Taylor [42], giving a total of fifteen methods. It is shown that the exponential smoothing family has good forecasting accuracy over several competitors [28–30], and it is especially suitable for short time series. There are three main types of exponential smoothing methods: (i) simple exponential smoothing (SES), that is suitable for forecasting time series data without any trend or seasonality; (ii) Holt's exponential smoothing, an extension of SES that incorporates trend information; (iii) Holt-Winters exponential smoothing, that extends Holt's method to incorporate seasonality in the data. Also, there are two variations of Holt-Winters method: additive and multiplicative. The ETS models can capture a variety of trend and seasonal structures (additive or multiplicative) and combinations of those. In order to refer to the three components error, trend, and seasonality in exponential smoothing methods; the notation ETS is proposed in [16] and we also use this notation. A detailed description of ETS can be found in [17] and is therefore not repeated here.

The selection of an appropriate exponential smoothing model and its parameters depends on the data characteristics and the desired level of smoothing. We apply the `ets` function from the `forecast`

package to find automatically the best ETS model. This function implement the innovative state space modeling framework described in [17] for parameter estimation and forecasting.

4.4 TBATS model

An innovations state space modeling framework has been introduced in [23] for forecasting complex seasonal time series such as those with multiple seasonal periods, high-frequency seasonality, non-integer seasonality, and dual-calendar effects. This model, which is called BATS, is an exponential smoothing state space model with Box-Cox transformation, ARMA errors, Trend and Seasonal components. This model is a generalization of the traditional seasonal innovative models to allow multiple seasonal periods. In TBATS model, the trigonometric representation of seasonal components based on Fourier transform is used and the initial T in the notation TBATS stands for trigonometric. For more information on the theory and applications of TBATS, see [23]. The `tbats` function is made available through the `forecast` package to fit TBATS model to a time series.

5 Real data study

The data used in this paper is obtained from the Regional Water Company of Guilan province in Iran (https://www.glrw.ir). It includes monthly rainfall data (in mm) from May 2015 to June 2023 at four rain gauge stations: (i) Hashtpar, (ii) Manjil, (iii) Rasht, and (iv) Shalman. In this section, we compare the classical SSA and the robust SSA algorithms with other forecasting methods, including NNAR, ARIMA, ETS, TBATS, in terms of accuracy for model fit and model forecast.

Table 1 shows the descriptive statistics of the monthly rainfall recorded in the four stations, including the minimum, maximum, mean, and standard deviation. The standard deviations of the original data are large, indicating that the monthly rainfall has dramatic fluctuations, and thus difficult for modeling. Manjil is the station that shows the smallest variations among the considered stations, and low mean rainfall. On the other end, there are Shalman and Rasht showing larger variations with higher mean monthly rainfall. In addition to the descriptive measures, Figure 1 shows the movements of the monthly rainfall at the four stations over time. Application of the `tsoutliers` function from the `forecast` package reveals the presence of two outlier data points in Manjil station and three in Shalman station. Conversely, no outlier data points are detected in Hashtpar and Rasht stations. Even a small proportion of outliers can significantly impact model fitting and forecasting results.

Table 1: Descriptive measures for the monthly rainfall at the four rain gauge stations.

Station	N	Minimum	Maximum	Mean	Standard deviation
Hashtpar	98	0.0	304.5	84.0102	66.0732
Manjil	98	0.0	112.0	21.6092	22.9365
Rasht	98	0.7	342.0	100.2449	78.5913
Shalman	98	0.0	477.5	96.3602	91.8995

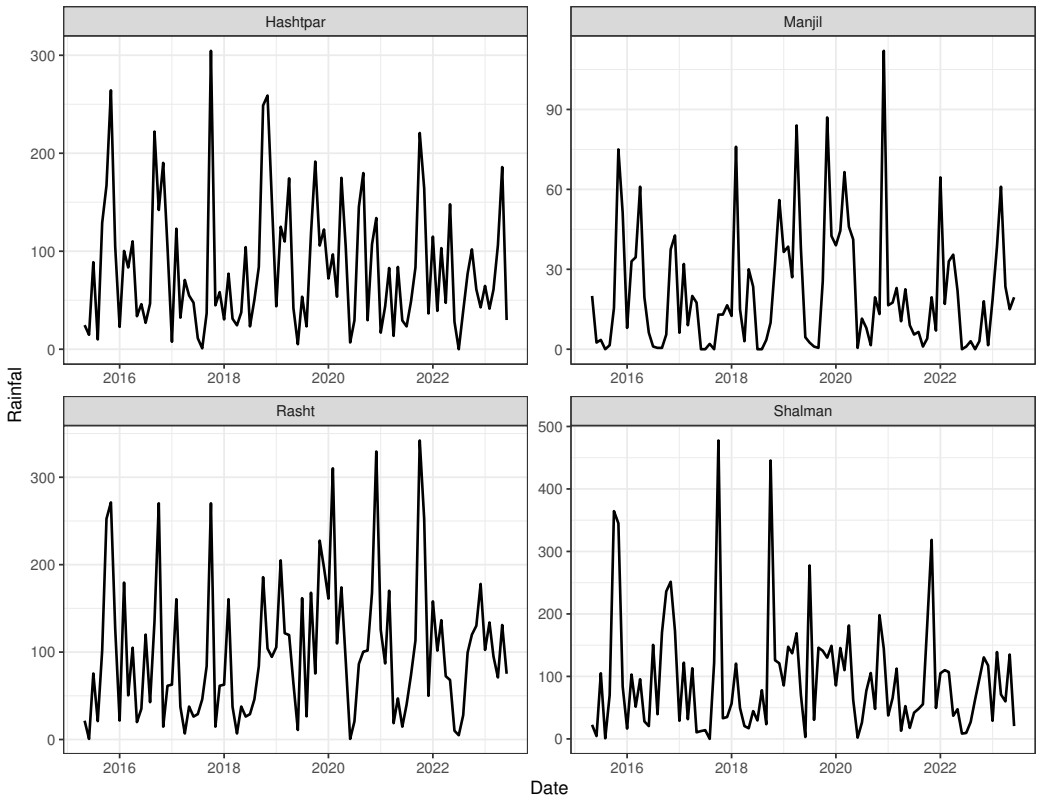

Figure 1: Time series for the monthly rainfall at the four rain gauge stations.

5.1 Model fit

As mentioned in the Subsection 2.3, for SSA and robust SSA algorithms, there are two choices to be made: (i) the window length L; and (ii) the number of eigentriples used for reconstruction r. The window length L could be approximately half of the time series length N or $\frac{1}{4}$ depending on the length of time series, and proportional to the number of observations per period (e.g. to 12 for monthly time series, to four for quarterly time series, etc.). Two values of L are chosen for each time series, $L_1 = 24$ and $L_2 = 48$ accordingly. The choice of the number of eigentriples used for reconstruction r, for each of the considered window lengths and each of the time series, is done by taking into consideration the w-correlations among components. Figure 2 shows the w-correlation matrices for each of the four stations, considering the window length $L = 24$. The results for $L = 48$ are similar, so we do not present them here for the sake of brevity. The w-correlation matrices can be obtained with the function `wcor` of the `R` package `Rssa` [6] and the number of eigentriples r should be chosen in order to maximize the separability between signal and noise components; i.e., maximize the w-correlation among signal components, maximize the w-correlation among noise components, and minimize the w-correlation between signal and noise components. Figure 2 indicates that the optimal number of eigentriples used for reconstruction is $r = 5$ for Hashtpar and Manjil stations, whereas for Rasht and Shalman stations, the optimal value is $r = 3$.

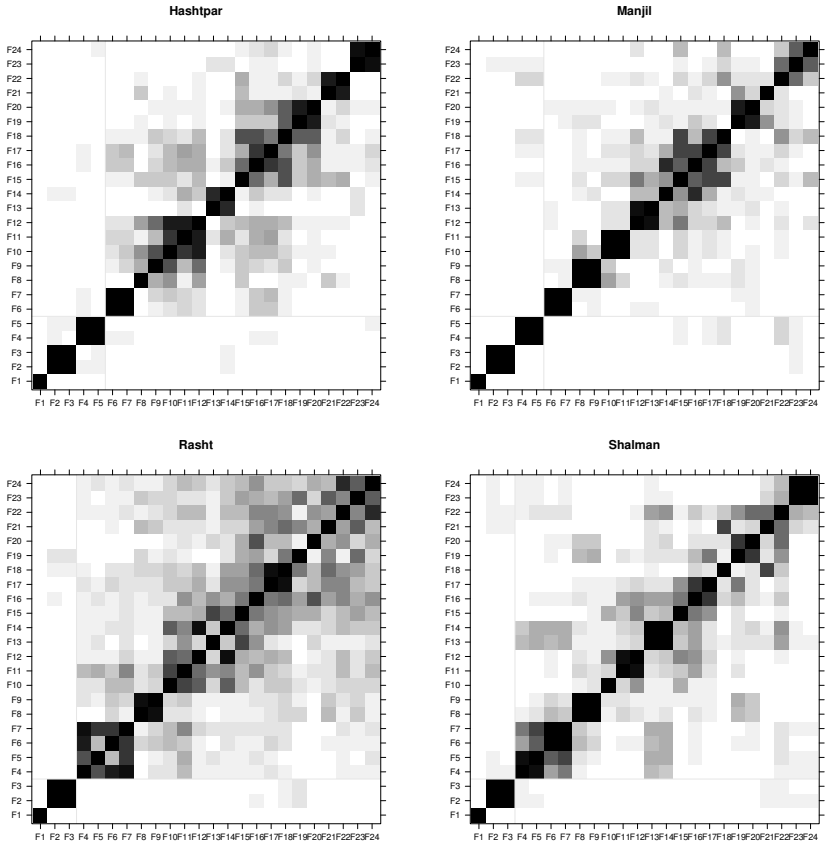

Figure 2: *w*-correlation matrices for each of the four rain gauge stations, considering the window length $L = 24$.

To identify the order of ARIMA model for the series, we followed the procedure outlined in [17]. This approach involves a combination of unit root tests, AICc minimization (AIC with a correction for finite sample sizes), and maximum likelihood estimation. The specific ARIMA models determined for the Hashtpar, Manjil, Rasht, and Shalman stations are ARIMA$(0,0,0)(2,0,0)_{12}$, ARIMA$(0,0,0)(0,1,1)_{12}$, ARIMA$(0,0,0)(1,0,0)_{12}$, and ARIMA$(0,0,0)(1,0,0)_{12}$, respectively. Figure 3 displays the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of the monthly rainfall for each station. Based on these plots, it might be argued that the ARIMA model fitted using the method in [17] may not be the best fit for these time series, and better models could potentially be identified. This is due to the fact that instead of exhaustively considering every possible combination of p and q, the algorithm employs a stepwise search to traverse the model space. If the fitted model accurately captures the trends, variability, and correlation structure present in the time series data, we expect the residuals of the model to be uncorrelated. To assess the presence of correlation in the residuals, ACF plot or the Ljung-Box test can be employed. Table 2 presents the results of Ljung-Box test to evaluate the presence of autcorrelation in the residuals of ARIMA fitted models across the four rain gauge stations. We also presented ACF plots of the residuals in Figure 4.

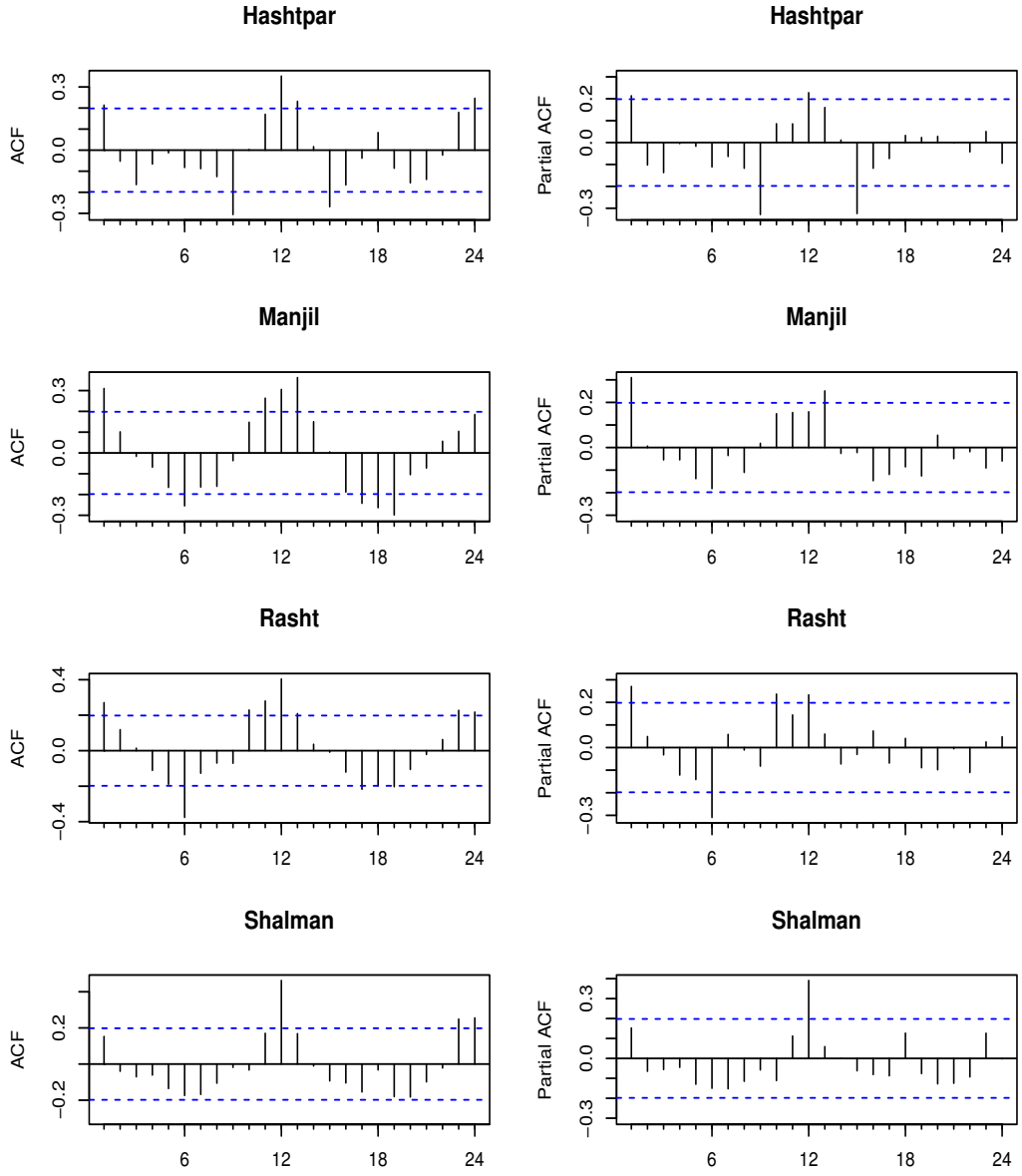

Figure 3: ACF (left) and PACF (right) plots for each of the four rain gauge stations, considering the window length $L = 24$.

From Figure 4, it is observed that for Hashtpar and Manjil stations, the ACF plots of the residuals show autocorrelations within the bounds of $\pm 1.96/\sqrt{n}$. This suggests that the autocorrelations of the residuals are not significant. However, for Rasht, a single spike is observed at lag 6, and for Shalman station, a single spike is observed at the lag 20. Nevertheless, based on the Ljung-Box test for residuals, all p-values are greater than $\alpha = 0.05$, indicating that autocorrelations in the residuals are not significant.

Table 2: Results of Ljung-Box test to evaluate the presence of autcorrelation in the residuals of ARIMA fitted models across the four rain gauge stations.

Station	ARIMA$(p,d,q)(P,D,Q)_m$	Q-statistic	p-value
Hashtpar	ARIMA$(0,0,0)(2,0,0)_{12}$	22.545	0.1647
Manjil	ARIMA$(0,0,0)(0,1,1)_{12}$	26.192	0.1249
Rasht	ARIMA$(0,0,0)(1,0,0)_{12}$	23.340	0.1779
Shalman	ARIMA$(0,0,0)(1,0,0)_{12}$	25.489	0.0842

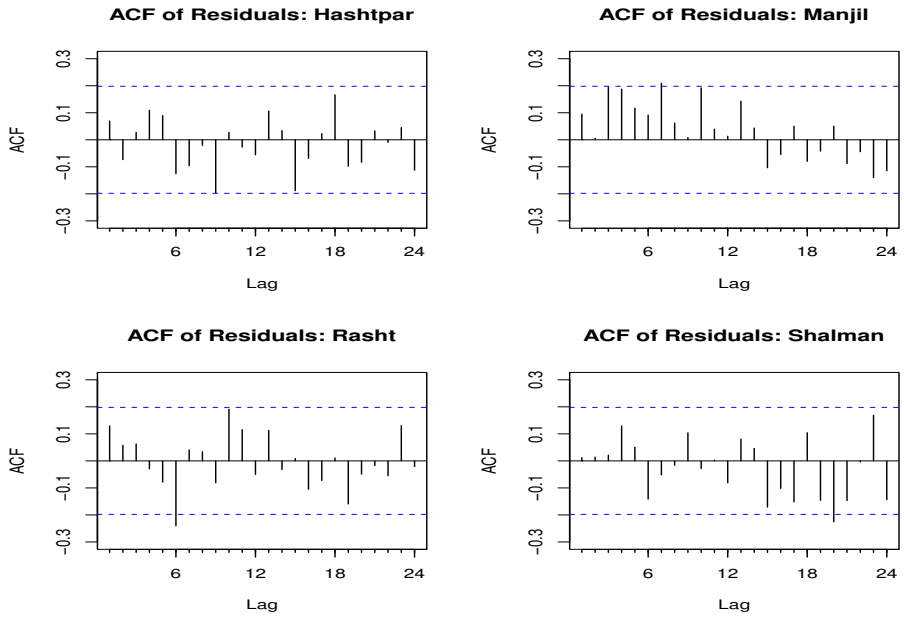

Figure 4: Acf plots for the residuals of ARIMA fitted models across the four rain gauge stations.

In order to evaluate and compare the ability for model fit using the seven models, SSA, robust SSA based on the L_1 norm (RL-SSA), and robust SSA based on the Huber function (RH-SSA), NNAR, ARIMA, ETS, and TBATS, the root mean square error (RMSE) is calculated for each time series as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \widetilde{y}_t)^2}, \tag{14}$$

where y_t are the observed values and $\widetilde{y}_t$ the fitted values by the considered model/algorithm.

Table 3 shows the RMSE for model fit by each of the seven models applied to each of the four stations, considering a window length $L_1 = 24$ and $L_2 = 48$. The bold numbers show the method with the lowest RMSE for a given station. From this table, we can conclude that when the window length is set to be 48, the classical SSA provides the best results, while the robust SSA algorithms has the second best performances. However, when the window length in the SSA related algorithms is set to be 24, NNAR

followed by TBATS outperforms the other methods for Rasht and Shalman stations, while classical SSA has the best performance for the other two stations. In summary, for the three last stations (Manjil, Rasht and Shalman), classical SSA with window length $L = 48$ provided lowest RMSE values for model fit, while robust SSA based on Huber function (RH-SSA) has the best performance for the Hashtpar station.

Table 3: RMSE results of all models for model fit (the bold numbers show the method with the lowest RMSE, and underlined numbers represent the second best method, for a given station).

Method	L	Hashtpar	Manjil	Rasht	Shalman
SSA	24	45.1356	<u>14.7954</u>	59.0636	74.7610
SSA	48	<u>38.0538</u>	**14.4480**	**42.5357**	**48.6417**
RL-SSA	24	51.6824	17.4490	64.5301	82.7758
RL-SSA	48	39.4138	16.5495	49.6822	60.0780
RH-SSA	24	45.5833	15.7424	60.5624	77.0145
RH-SSA	48	**37.1327**	15.4243	<u>44.5721</u>	<u>55.5056</u>
ARIMA	-	59.01134	21.2334	69.9947	77.80322
NNAR	-	56.14799	16.9339	53.5626	60.1228
ETS	-	49.97855	17.44423	61.45218	66.61744
TBATS	-	49.99718	17.06914	58.48833	65.52479

5.2 Model forecasting

In this subsection, we compare the forecasting abilities of SSA, robust SSA based on the L_1 norm (RL-SSA), robust SSA based on the Huber function (RH-SSA), and other forecasting methods including ARIMA, NNAR, ETS, and TBATS. The RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{g}\sum_{t=N_0+h+1}^{N}(y_t - \widetilde{y}_t)^2},$$

where $N_0 = N - h - g$, y_t are the last g observed values used as the testing set, and $\widetilde{y}_t$ the respective h steps-ahead forecast values. The RMSE is sensitive even to small errors, which can size the model performance for high rainfall values. Table 4 shows the RMSE for the monthly rainfall forecasting at each of the four rain gauge stations, considering each of the seven models. In this table, the bold font shows the forecasting method with the lowest RMSE for a given station. The data from May 2015 to June 2022 with a total of 86 observations from each time series are used for estimation purposes as the training data. The remainder with 12 observations are used to evaluate the forecasting performance of the models as the testing set. One, five, and ten steps ahead out-of-sample forecast is performed to assess the prediction performance. Figure 5 shows the original time series (in black) along with one-month-ahead forecasts for the testing data, obtained from the seven models, across the four rain gauge stations. Additionally, Figure 6 displays the original time series (in black) alongside the predicted values for the training data.

Table 4: RMSE results of all models for forecasting (the bold numbers show the method with the lowest RMSE, and underlined numbers represent the second best method, for a given station).

Station	SSA	RL-SSA	RH-SSA	ARIMA	NNAR	ETS	TBATS
			one-step-ahead				
Hashtpar	45.1631	**41.8713**	44.3003	46.6307	65.4170	49.1724	45.0945
Manjil	13.8279	15.1882	**12.1176**	15.4223	14.3575	18.2529	15.1371
Rasht	57.0732	43.8261	50.3878	**39.6891**	66.0762	45.2730	40.8733
Shalman	43.7358	56.8805	**41.8069**	52.1285	93.3484	51.4315	50.6261
			five-step-ahead				
Hashtpar	46.3739	49.1631	**43.4417**	49.1971	58.0938	49.2797	44.5486
Manjil	14.1279	15.1702	**12.3004**	18.4389	21.7046	18.8373	18.3192
Rasht	67.2449	47.0623	59.9564	**46.6603**	67.2273	111.8616	51.8536
Shalman	44.4978	54.3320	**42.7679**	52.7647	81.3589	52.2147	50.1439
			ten-step-ahead				
Hashtpar	47.6808	49.9914	45.2679	49.2458	67.0889	49.2116	**42.4314**
Manjil	14.6204	**11.9890**	12.3174	17.9994	22.4835	17.8495	16.5358
Rasht	73.1706	51.8973	72.4721	**44.9188**	96.7256	195.8318	45.2398
Shalman	43.1522	63.6617	**40.1460**	51.0000	47.0983	51.8971	50.5451

From Table 4, it can be observed that robust SSA algorithm based on the Huber function (RH-SSA) achieved the best performance in terms of RMSE for one-step-ahead forecasting in the two considered time series. Conversely, robust SSA algorithm based on the L_1 norm and ARIMA model demonstrated the best performance in one of the time series in this case. Furthermore, RH-SSA model exhibited the best performance for five-step-ahead forecasting in the three time series. Interestingly, ARIMA only yielded the best performance in the Rasht gauge station. Additionally, for ten-step-ahead forecasting, each of the RH-SSA, RL-SSA, ARIMA, and TBATS models achieved the best performance in one particular case. In summary, the monthly rainfall forecasting at the Hashtpar, Manjil, and Shalman stations, robust RH-SSA algorithm emerged as the superior model followed by RL-SSA and classical SSA. Similarly, for the Rasht station, ARIMA proved to be the best model followed by TBATS. Moreover, in most instances, TBATS exhibited better performance compared to ARIMA, NNAR, and ETS; however, it was consistently outperformed by SSA-related methods. Figure 5 also confirms the above findings. The NNAR model, as it utilizes lagged values as inputs, lacks predicted values for some initial observations. Consequently, in Figure 6, the fitted values for the initial points are not shown. Figure 6 clearly illustrates that NNAR performs very well for the training data at Hashtpar Station. However, as depicted in Figure 5, this approach demonstrates poor performance in forecasting the testing data in Hashtpar, indicating overfitting. This holds true for TBATS at Hashtpar and Shalman stations as well. Its performance is satisfactory for the training data, but not suitable for forecasting the test data.

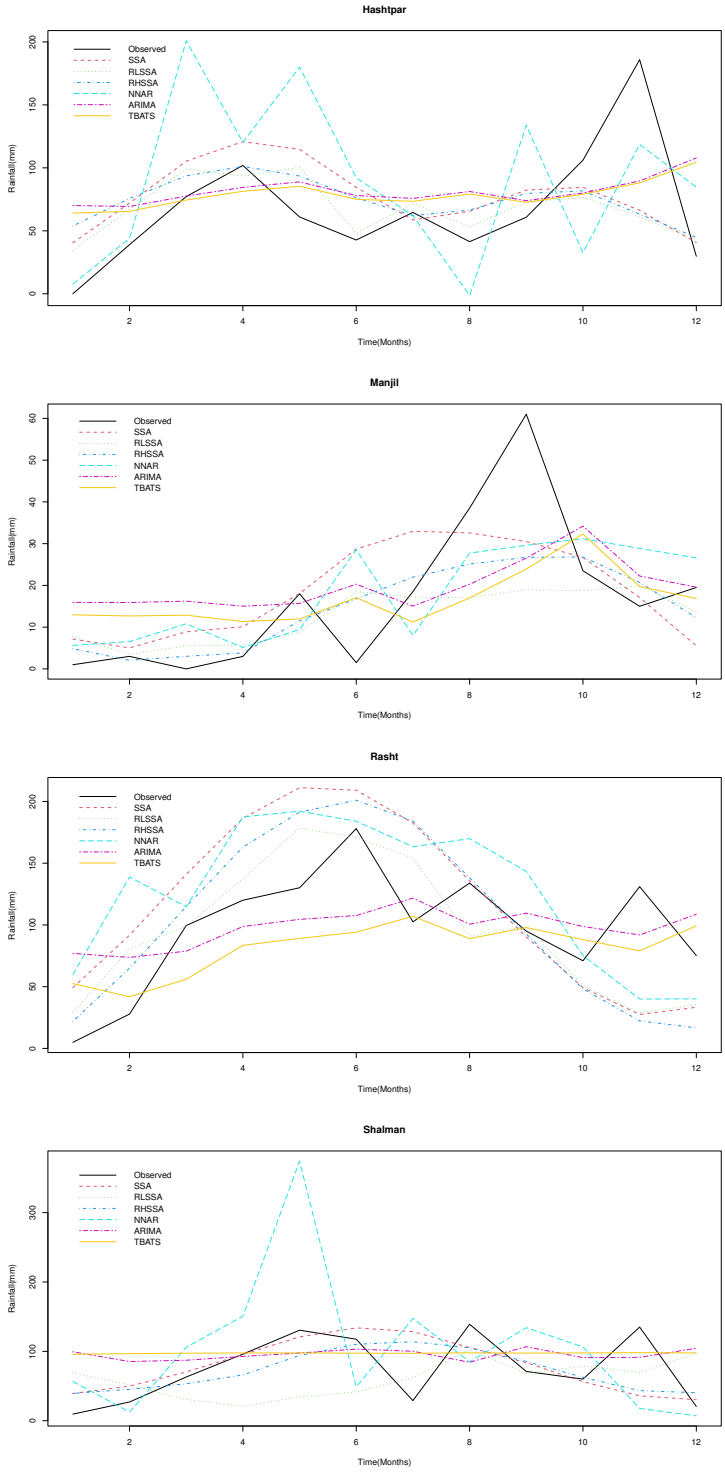

Figure 5: Forecasting results of the testing data across the four rain gauge stations.

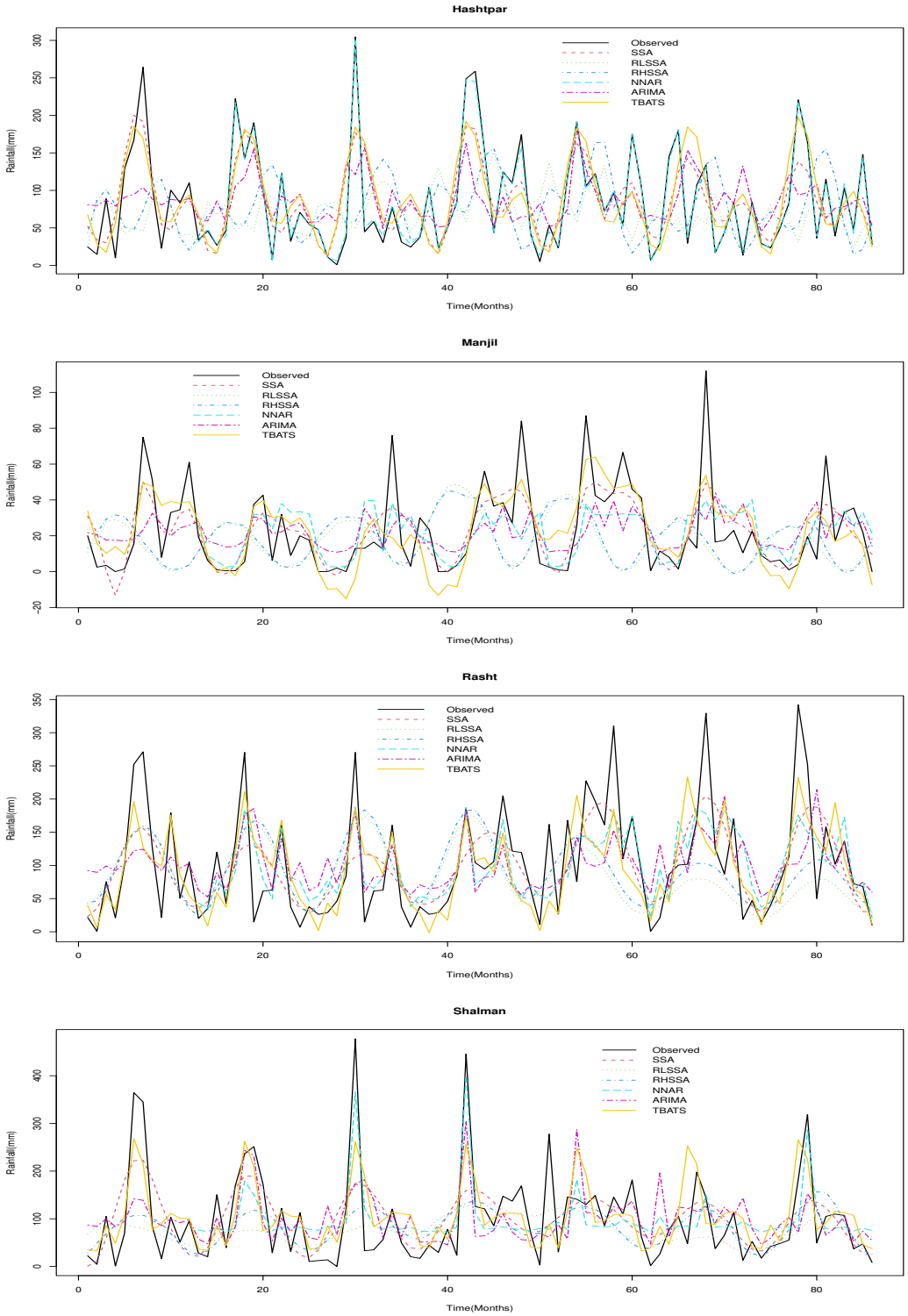

Figure 6: Predicted values for the training data across the four rain gauge stations.

6 Conclusions

In this paper, we utilized the classical SSA methodology as well as two robust SSA algorithms, RL-SSA and RH-SSA, to achieve model fit and forecast the monthly rainfall. To evaluate the effectiveness of these approaches, we compared the results with those obtained from other commonly used time series forecasting techniques, such as NNAR, ARIMA, ETS, and TBATS, using the RMSE criterion.

Our study focused on four rain gauge stations in Guilan province, specifically Hashtpar, Manjil, Rasht, and Shalman. The evidence gathered from this investigation demonstrates that there is no single model to be the best for any of the stations. We expect that our research will serve as a useful tutorial for government entities in selecting an appropriate model for rainfall forecasting. Such decisions are pivotal in effectively predicting flash floods and managing water resources.

Acknowledgements

The author would like to thank the editor and anonymous reviewers for providing helpful suggestions which contributed to the improvement of the paper.

References

[1] F.R. Adaryani, S. Jamshid Mousavi, F. Jafari, *Short-term rainfall forecasting using machine learning-based approaches of PSO-SVR, LSTM and CNN*, J. Hydrol. **614** (2022) 128463.

[2] G.E. Box, G.M. Jenkins, *Time Series Analysis- Forecasting and Control*, Colarado State University, Holden day, San Francisco, 1976.

[3] P.J. Brockwell, R.A. Davis, *Introduction to Time Series and Forecasting*, Springer, New York, 1996.

[4] D.S. Broomhead, G.P. King, *Extracting qualitative dynamics from experimental data*, Phys. D Nonlinear Phenom. **20** (1986) 217–236.

[5] K.R. Gabriel, S. Zamir, *Lower rank approximation of matrices by least squares with any choice of weights*, Technometrics **21** (1979) 489–498.

[6] N. Golyandina, A. Korobeynikov, A.Shlemov, K. Usevich, *Multivariate and 2D extensions of singular spectrum analysis with the Rssa package*, J. Stat. Softw. **67** (2015) 1–78.

[7] N. Golyandina, V. Nekrutkin, A. Zhigljavsky, *Analysis of Time Series Structure: SSA and Related Techniques*, Chapman & Hall/CRC, New York, 2001.

[8] N. Golyandina, A. Zhigljavsky, *Singular Spectrum Analysis for Time Series*, Springer, Berlin, Heidelberg, 2013.

[9] N. Golyandina, A. Korobeynikov, A. Zhigljavsky, *Singular Spectrum Analysis with R*, Springer, Berlin, Heidelberg, 2018.

[10] A. Groth, M. Ghil, *Synchronization of world economic activity*, Chaos: An Interdisciplinary. J. Nonlinear Sci. **27** (2017) 127002.

[11] H. Hassani, R. Mahmoudvand, H.N. Omer, E.S. Silva, *A preliminary investigation into the effect of outlier(s) on singular spectrum analysis*, Fluct. Noise Lett. **13** (2014) 1450029.

[12] H. Hassani, R. Mahmoudvand, *Multivariate singular spectrum analysis: A general view and new vector forecasting approach*, Int. J. Energy Res. Stat. **1** (2013) 55–83.

[13] D.M. Hawkins, L. Liu, S. Young, *Robust singular value decomposition*, Natl. Inst. Stat. Sci. **122** (2001) 1–12.

[14] P.J. Huber, *Robust Statistics*, Wiley, New York, 1981.

[15] R. J. Hyndman, G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. OHara-Wild, F. Petropoulos, S. Razbash, E. Wang, F. Yasmeen, *Forecast: Forecasting functions for time series and linear models, R package version 8.15*, 2021. Available online: pkg.robjhyndman.com/forecast/.

[16] R.J. Hyndman, A.B. Koehler, J.K. Ord, R.D. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*, Springer, Berlin, Heidelberg, 2008.

[17] R. Hyndman, Y. Khandakar, *Automatic time series forecasting: The forecast package for R*, J. Stat. Softw. **27** (2008) 1–22.

[18] R.J. Hyndman, G. Athanasopoulos, *Forecasting: Principles and Practice, 2nd Edition*, OTexts, Melbourne, 2018.

[19] R.J. Hyndman, A.B. Koehler, R.D. Snyder, S. Grose, *A state space framework for automatic forecasting using exponential smoothing methods*, Int. J. Forecast. **18** (2002) 439–454.

[20] M. Kazemi, P.C. Rodrigues, *Robust singular spectrum analysis: comparison between classical and robust approaches for model fit and forecasting*, Comput. Stat. (2023) 1-33. doi: 10.1007/s00180-022-01322-4.

[21] M.A.R. Khan, D.S. Poskitt, *A note on window length selection in singular spectrum analysis*, Aust. N. Z. J. Stat. **55** (2013) 87–108.

[22] N. Kumar, M. Nasser, S.C. Sarker, *A new singular value decomposition based robust graphical clustering technique and its application in climatic data*, J. Geol. Geogr. Geoecology **3** (2011) 227–238.

[23] A.M.D. Livera, R.J. Hyndman, R.D. Snyder, *Forecasting time series with complex seasonal patterns using exponential smoothing*, JASA **106** (2011) 1513–1527.

[24] D. Machiwal, M.K. Jha, *Hydrologic Time Series Analysis: Theory and Practice*, Springer, Capital Publishing company, 2012.

[25] R. Mahmoudvand, P.C. Rodrigues, *A new parsimonious recurrent forecasting model in singular spectrum analysis*, J. Forecast. **37** (2018) 191–200.

[26] R. Mahmoudvand, N. Najari, M. Zokaei, *On the optimal parameters for reconstruction and forecasting in singular spectrum analysis*, Commun. Stat. Simul. Comput. **42** (2013) 860–870.

[27] R. Mahmoudvand, P.C. Rodrigues, M. Yarmohammadi, *Forecasting daily exchange rates: A comparison between SSA and MSSA*, RevStat-Stat. J. **17** (2019) 599–616.

[28] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, R. Winkler, *The accuracy of extrapolation (time series) methods: Results of a forecasting competition*, J. Forecast. **1** (1982) 111–153.

[29] S. Makridakis, M. Hibon, *The M3-Competition: results, conclusions and implications*, Int. J. Forecast. **16** (2000) 451–476.

[30] S. Makridakis, E. Spiliotis, V. Assimakopoulos, *The M4-Competition:* 100,000 *time series and* 61 *forecasting methods*, Int. J. Forecast. **36** (2020) 54–74.

[31] C.C. Pegels, *Exponential forecasting: Some new variations*, Manag. Sci. **15** (1969) 311–315.

[32] M.C. Ramírez, N.J. Ferreira, H.F. Velho, *Linear and nonlinear statistical downscaling for rainfall forecasting over southeastern Brazil*, Weather **21** (2006) 969–989.

[33] M.U.M. Rao, K.C. Patra, S.K. Sasmal, A. Sharma, G. Oliveto, *Forecasting of rainfall across river basins using soft computing techniques: the case study of the upper Brahmani basin (India)*, Water **15** (2023) 499.

[34] V.A. Reisen, F.F. Molinares, *Robust estimation in time series with long and short memory properties*, Ann. Math. Inform. **39** (2012) 207–224.

[35] P.C. Rodrigues, A. Monteiro, V.M. Loureno, *A Robust additive main effects and multiplicative interaction model for the analysis of genotype-by-environment data*, Bioinform. **32** (2016) 58–66.

[36] P.C. Rodrigues, R. Mahmoudvand, *Correlation analysis in contaminated data by singular spectrum analysis*, Qual. Reliab. Eng. Int. **32** (2016) 2127–2137.

[37] P.C. Rodrigues, J. Pimentel, P. Messala, M. Kazemi, *The decomposition and forecasting of mutual investment funds using singular spectrum analysis*, Entropy **22** (2020) 83.

[38] P.C. Rodrigues, V.M. Lourenco, R. Mahmoudvand, *A robust approach to singular spectrum analysis*, Qual. Reliab. Eng. Int. **34** (2018) 1437–1447.

[39] P.C. Rodrigues, R. Mahmoudvand, *A new approach for the vector forecast algorithm in singular spectrum analysis*, Commun. Stat. Simul. Comput. **49** (2020) 591–605.

[40] M. Shahin, H.J. Van Oorschot, S.J. De Lange, *Statistical Analysis in Water Resources Engineering*, Aa Balkema, Rotterdam, 1993.

[41] W. Sulandari, M.H. Lee, P.C. Rodrigues, *Indonesian electricity load forecasting using singular spectrum analysis*, Energy **190** (2020) 116408.

[42] J.W. Taylor, *Exponential smoothing with a damped multiplicative trend*, Int. J. Forecast. **19** (2003) 715–725.

[43] A. Veloz, R. Salas, H. Allende-Cid, H. Allende, C. Moraga, *Identification of lags in nonlinear autoregressive time series using a flexible fuzzy model*, Neural Process. Lett. **43** (2016) 641–666.

[44] R. Wang, H. Ma, G. Liu, D. Zuo, *Selection of window length for singular spectrum analysis*, J. Frank. Inst. **352** (2015) 1541–1560.

[45] L. Zhang, H. Shen, J.Z. Huang, *Robust regularized singular value decomposition with application to mortality data*, Ann. Appl. Stat. **7** (2013) 1540–1561.

[46] J. Zabalza, C. Qing, P. Yuen, G. Sun, H. Zhao, J. Ren, *Fast implementation of two-dimensional singular spectrum analysis for effective data classification in hyperspectral imaging*, J. Frankl. Inst. **355** (2018) 1733–1751.