

Article

A Developed Multiple Linear Regression (MLR) Model for Monthly Groundwater Level Prediction

Mohammad Ehteram ^{1,*} and Fatemeh Barzegari Banadkooki ²

¹ Department of Water Engineering and Hydraulic Structures, Faculty of Civil Engineering, Semnan University, Semnan 35131-19111, Iran

² Agricultural Department, Payame Noor University, Tehran 19395-4697, Iran

* Correspondence: mohammdehteram@semnan.ac.ir

Abstract: Groundwater level (GLW) prediction is essential for monitoring water resources. Our study introduces a novel model called convolutional neural network (CNN)–long short-term memory neural network (LSTM)–Multiple linear regression (MLR) for groundwater level prediction. We combine two deep learning models with the MLR model to predict GWL and overcome the limitations of the MLR model. The current paper has several innovations. Our study develops an advanced hybrid model for predicting groundwater levels (GWLs). The study also presents a novel feature selection method for selecting optimal input scenarios. Finally, an advanced method is developed to examine the impact of inputs and model parameters on output uncertainty. The current paper introduces the gannet optimization algorithm (GOA) for choosing the optimal input scenario. A CNN-LSTM-MLR model (CLM), CNN, LSTM, MLR model, CNN-MLR model (CNM), LSTM-MLR model (LSM), and CNN-LSTM model (CNL) were built to predict one-month-ahead GWLs using climate data and lagged GWL data. Output uncertainty was also decomposed into parameter uncertainty (PU) and input uncertainty (IU) using the analysis of variance (ANOVA) method. Based on our findings, the CLM model can successfully predict GWLs, reduce the uncertainty of CNN, LSTM, and MLR models, and extract spatial and temporal features. Based on the study's findings, the combination of linear models and deep learning models can improve the performance of linear models in predicting outcomes. The GOA method can also contribute to feature selection and input selection. The study findings indicated that the CLM model improved the training Nash–Sutcliffe efficiency coefficient (NSE) of the CNL, LSM, CNM, LSTM, CNN, and MLR models by 6.12%, 9.12%, 12%, 18%, 22%, and 30%, respectively. The width intervals (WIs) of the CLM, CNL, LSM, and CNM models were 0.03, 0.04, 0.07, and 0.12, respectively, based on IU. The WIs of the CLM, CNL, LSM, and CNM models were 0.05, 0.06, 0.09, and 0.14, respectively, based on PU. Our study proposes the CLM model as a reliable model for predicting GWLs in different basins.



Citation: Ehteram, M.; Banadkooki, F.B. A Developed Multiple Linear Regression (MLR) Model for Monthly Groundwater Level Prediction. *Water* **2023**, *15*, 3940. <https://doi.org/10.3390/w15223940>

Academic Editor: Alexander Yakirevich

Received: 18 October 2023

Revised: 3 November 2023

Accepted: 4 November 2023

Published: 11 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: deep learning models; feature selection models; groundwater level; hybrid models

1. Introduction

Groundwater plays a key role in managing water demands [1]. Groundwater is a valuable resource for reducing droughts. Groundwater level prediction is essential for sustainable water resource management [2]. Predicting groundwater levels can be used to monitor water availability during droughts [3]. Coastal areas may experience saltwater intrusion into freshwater aquifers due to excessive groundwater extraction.

Groundwater level (GWL) prediction is necessary to prevent saltwater intrusion into freshwater aquifers [3,4]. GWL prediction is complex because it is affected by many variables, including human activities and climatic conditions [5]. Human activities can affect groundwater levels. Climate parameters also influence the GWL [5]. Variables such as precipitation, temperature, humidity, and evaporation can alter groundwater levels [6].

Machine learning and numerical models can be used to predict the GWL. There are different packages and numerical models for predicting the GWL. For example, the modular groundwater flow model is a mathematical groundwater model that can predict the GWL. Numerical models have complex equations and boundary conditions [7]. The complexity of numerical models can limit their interpretation [8].

A machine learning model can predict a variety of hydrological variables [7]. Machine learning models (MLM) can capture nonlinear components of time series data [8]. Since MLM models use different layers and operators, they can accurately predict the GWL [8]. Hydrologists look for simple and accurate models to predict hydrological variables [8].

Multiple linear regression (MLR) is a simple and accurate machine learning model. The MLR model provides a linear relationship between inputs and outputs [9]. Since the MLR model is user-friendly, scholars use it for hydrological predictions [9]. Table 1 displays a review of studies implementing the MLR model for GWL prediction.

Table 1. Summary of studies employing multiple linear regression for Groundwater Level Prediction.

| References | Results | Discussion |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sahoo and Jha [10] | Developed the MLR model to predict Groundwater Level (GWL). Rainfall, river stage, and temperature were used to predict GWL. They tested different input combinations for predicting GWL. They reported that the MLR model was a reliable tool for groundwater modeling | For predicting GWL, the paper used the original SVM and MLR models. However, the original versions of these models may not fully extract important features and accurately predict GWL. Furthermore, random input selection may not result in high accuracy. |
| Ebrahimi and Rajaei [11] | Ebrahimi and Rajaei [11] coupled the wavelet method with the MLR, artificial neural network models (ANNs), and support vector machine models (SVMs) to predict GWL. They used the wavelet method to decompose time series into multiple subtime series. Compared to other wavelets, the Db5 wavelets performed better. The wavelet-MLR model improved the precision of the MLR model for predicting GW. | Wavelet preprocessed data points. Wavelet transforms can be used to extract valuable features from time series data. These features allow predictive models to capture both high-frequency and low-frequency components. |
| Bahmani and Ouarda [12] | Bahmani and Ouarda [12] used the decision-tree model, the genetic programming (GEP) model, and the MLR model for predicting Groundwater Level (GWL). The Ensemble Empirical Mode Decomposition (EEMD) was used to preprocess the data. Wavelet-GEP performed better than other models. | Since the MLR model lacked advanced operators for analyzing nonlinear data, its accuracy was lower than that of the GEP and EEMD-GEP models. EEMD decomposes the original signal into IMFs that capture more detailed and local features. |
| Poursaeid et al. [13] | Poursaeid et al. [13] used the MLR, the extreme learning machine model (ELM), and the SVM model to predict GWL. The study concluded that the ELM model performed better than the MLR model. | Due to the lack of robust optimizers, the model did not achieve high accuracy. To improve the performance of the ELM, SVM, and MLR models, robust optimizers were needed. |

The MLR model has been widely used to predict GWLs, but its main limitations have not been addressed. Studies have failed to improve the performance of the MLR model for analyzing nonlinear data. While GWL data exhibit spatial and temporal variations, MLR models have not been developed to extract spatial and temporal features. The main problem is that MLR assumptions cannot handle GWL data and nonlinear relationships.

The MLR model does not capture nonlinear components of time series data, so it cannot accurately predict the GWL [10]. MLR uses linear assumptions to predict the GWL, but inputs and outputs may have intricate and nonlinear relationships [13]. By improving the efficiency of the MLR model, we can ensure that it captures the underlying relationships between variables more effectively.

Additionally, MLR cannot effectively handle output uncertainty. Furthermore, the performance of the MLR model depends on the correct selection of inputs [10]. In recent

years, researchers have used hybrid models to overcome the limitations of classical machine learning models [14]. By combining multiple models, a hybrid model improves prediction accuracy and overcomes the shortcomings of traditional models [14]. We present an innovative model for predicting GWLs that addresses the limitations of the MLR model. This new model improves the ability of the MLR model to capture nonlinear components, quantify uncertainty values, and determine the best input scenarios.

Since linear models such as MLR cannot fully extract important features from GWL data [10], our article focuses on developing robust models to predict GWLs spatially and temporally. To overcome the drawbacks of linear models, this article combines two robust deep learning models with MLR to predict GWLs spatially and temporally. Since deep learning models use advanced operators to extract spatial and temporal features, we choose them to improve the performance of the MLR model for extracting features and predicting data.

A deep learning model consists of multiple layers that analyze and predict data with high accuracy [4]. A convolutional neural network (CNN) is a deep learning algorithm that extracts valuable features from time series data [15]. The CNN model is a reliable tool for spatially analyzing GWL data and extracting important patterns [16]. The long short-term memory (LSTM) neural network is another robust tool for temporal analysis of groundwater level (GWL) data [17]. CNN and LSTM models can be used to improve the accuracy of the MLR model by analyzing and predicting nonlinear and complex data [15,16].

Ghasemlounia et al. [18] reported that different variants of the LSTM model could improve the accuracy of GWL predictions. They indicated that the bidirectional LSTM model could successfully predict the GWL. Sheikh Khozani et al. [19] stated that LSTM models were robust tools for improving the accuracy of linear models. A linear-LSTM model was used to predict groundwater levels (GWL) and the hybrid models enhanced the linear models' performance. Afshari Nia et al. [14] stated that the CNN model played a key role in extracting spatial features from hydrological time series data.

Our study employs a new model called convolutional neural network–long short-term memory neural network–MLR (CLM) to predict GWLs. The new method uses CNNs and LSTMs to analyze GWL data spatially and temporally. As a result, the hybrid structure of the new model can improve the predictive capability of the MLR model. A hybrid model improves the ability of the MLR model to analyze nonlinear time series data. This paper also proposes an efficient method for quantifying output uncertainty, which is necessary for evaluating the accuracy of models.

To select the optimal input scenario for groundwater level prediction, the current study proposes a novel feature selection method and compares it with other methods. The main novelties of this paper are the development of a new predictive model for groundwater level prediction and the introduction of a new method for selecting the best input scenarios.

The current paper includes the following novelties:

- An advanced machine learning model is developed to predict the monthly GWL. We combine three models to predict monthly groundwater levels. As our proposed model consists of two robust deep learning models, it can be superior to linear models such as MLR models. Since our proposed model uses the capabilities of CNN and LSTM models simultaneously, it outperforms standalone CNN and LSTM models. The CLM model is presented as a reliable tool for predicting GWLs in different basins. Since it can be adapted to different watersheds, it can be a reliable tool for water resource managers around the world.
- A new optimizer is introduced to identify the best input combination. An optimization algorithm determines the best input combination, while the correlation method only determines important inputs. To identify the most optimal set of features, the new method performs an extensive search and explores various combinations. The new optimization method handles high-dimensional feature spaces and can be fine-tuned to maximize model accuracy, minimize error, or enhance specific evaluation metrics. As a result, our study contributes to feature selection.

- An effective method is suggested to quantify uncertainty values. This study contributes to the quantification of output uncertainty values. The study illustrates how model parameters and inputs contribute to model uncertainties. Through the analysis of variance (ANOVA) method, the study provides insight into the sources and values of output uncertainty.
- We use an advanced method to decompose the output uncertainty into different sources of uncertainty. Our study proposes a new method to calculate the percentage contribution of inputs and model parameters to overall uncertainty.

This research contributes to water resource management by providing a reliable model for predicting groundwater levels. It addresses the limitations of traditional methods, introduces innovative techniques, and provides a comprehensive analysis of uncertainty and model performance.

The second section of the current study presents materials and methods. The third, fourth, and fifth sections present the case study, result, and the discussion, respectively.

2. Materials and Methods

2.1. MLR Model

The MLR model is a linear regression model that uses regression coefficients to determine the relationship between inputs and outputs [20]. The MLR model assumes a linear relationship between input and output data. An MLR model optimizes regression coefficients by minimizing prediction error. Equation (1) is used to provide outputs for the MLR model [20]:

$$Out = \beta_0 + \beta_1 X_1 \dots + \beta_n X_n \quad (1)$$

where *Out*: Output of variable X_1 : input variable, X_n : n th input variable, β_0 , β_1 and β_n : regression coefficients. As the MLR model does not use advanced operators, it may not be able to predict complex phenomena accurately. Since the original MLR model was unable to analyze nonlinear data, our study developed the MLR model to overcome these limitations.

2.2. Mathematical Model of the LSTM

The LSTM is a type of recurrent neural network that utilizes information gates and advanced structures to predict outputs [21]. The LSTM automatically extracts relevant features from sequences without the need for manual feature engineering. The input gate is one of the most important components of an LSTM model [22]. An input gate controls the flow of information. The current input data and the previous hidden state are processed using the TANH (hyperbolic tangent) activation function. LSTM models use forget gates to remember and forget information at each time step [23]. Output gates monitor the flow of information between memory cells and model outputs. Equations (2)–(6) provide outputs.

$$fo_u = \text{sigmoid}(\rho_f in_t + \rho_{f_h} h_{t-1} + b_f) \quad (2)$$

$$i_u = \text{sigmoid}(\rho_i in_t + \rho_{hi} h_{t-1} + b_f) \quad (3)$$

$$o_u = \text{sigmoid}(\rho_o in_t + \rho_{ho} h_{t-1} + b_0) \quad (4)$$

$$c_t = c_{t-1} \otimes (fo_u) + (i_u)_t \otimes (\tanh(\rho_c in_t + \rho_{hc} h_{t-1} + b_c)) \quad (5)$$

$$h_t = o_t \otimes \tanh(c_{t-1}) \quad (6)$$

where fo_u : output of forget gate; ρ_f and ρ_{f_h} : weight coefficients of forget gate; i_u : output of the input gate; in_t : input, b_c , b_f and b_0 : bias values; ρ_i and ρ_{hi} : weight coefficients of the input gate; *sigmoid*: an activation function; h_{t-1} : previous hidden state; ρ_o and ρ_{ho} : weight coefficients of the output gate; ρ_c and ρ_{hc} : weight coefficients of the cell state; c_t : new state of the cell state; o_u : output of the output gate; c_{t-1} : previous state of the cell state; \otimes : the element-wise multiplication; and h_t : the hidden state. Our research applies a robust algorithm to determine the best values of the LSTM parameters.

2.3. Mathematical Model of the CNN Model

The CNN model extracts spatial features from time series data [24]. CNNs use convolutional layers to automatically learn hierarchical features from input data. These layers consist of small filters (also called kernels) that slide over the input data and extract spatial features [25]. Convolutional layers produce feature maps. The pooling layer is the next layer of the CNN model. The pooling layer reduces the spatial dimension of feature matrices. The fully connected layer is the final layer of the CNN model. The average pooling method reduces the dimension of feature maps [26].

CNNs usually have one or more fully connected layers (also called dense layers) that perform classification or regression tasks [15]. Before applying the LSTM model, CNN output should be flattened into a one-dimensional vector [24].

2.4. Input Selection

Different climate parameters can change GWLs, so choosing the right input combination is important. Binary optimization algorithms are powerful tools for selecting the best input scenario. The algorithms can find the optimal input scenario by accurately searching the problem space [27]. Correlation coefficients only identify key input parameters, while binary algorithms automatically identify the best input combination. The Gannet optimization algorithm (GOA) is a novel algorithm for solving optimization problems. It was inspired by the life of Gannets. Pan et al. [28] reported that GOA was capable of solving mathematical functions and engineering problems. The GOA method converged faster than other algorithms. Our study selected this method because it provided promising results. Binary and continuous versions of the GOA were used to determine the LSTM and CNN parameters.

2.4.1. Structure of GOA

Gannets are fat birds with slender necks. A gannet is a seabird that resides in cold-temperate regions and uses unique methods to catch fish [28]. Gannets will form a straight line to catch fish. The gannet dives below the surface of the water to catch fish [28]. First, the initial position of the gannets is initialized:

$$GH_{ij} = r_1 \times (UB_j - LB_j) + LB_j \quad (7)$$

where GH_{ij} : location of the i th gannet in the j th dimension, UB_j and LB_j : upper and lower bounds of the decision variable, respectively; and r_1 : random value. Gannets follow prey using two types of diving patterns. Gannets use the U-shaped dive to reach deeper depths. Gannets use V-shaped dives to rapidly enter and exit the water [28]. Equations (8)–(10) are used to simulate these dive patterns. Equations (9) and (10) define the controller parameters. Equation (11) also defines a function based on the V-shaped dive to update the location of gannets.

$$\chi = 1 - \frac{IT}{IT_{\max}} \quad (8)$$

$$a = 2 \times \cos(2 \times \pi \times r_2) \times \chi \quad (9)$$

$$b = 2 \times V(2 \times \pi \times r_3) \times \chi \quad (10)$$

$$V(\phi) = \begin{cases} -\frac{1}{\pi} \times \phi + 1, & \phi \in (0, \pi) \\ \frac{1}{\pi} \times \phi - 1, & \phi \in (\pi, 2\pi) \end{cases} \quad (11)$$

where ϕ : the current solution; a and b : controller parameters; $V(\phi)$: the V-shaped dive; IT : iteration number; IT_{\max} : maximum number of iterations; and r_2 and r_3 : random numbers. Equations (13)–(16) are used to define controller parameters for updating the location of gannets. We choose one of the diving strategies to update the agent's location.

$$MGH_i(t + 1) = \begin{cases} GH_i(t) + m_1 + m_2, q \geq 0.50 \\ GH_i(t) + n_2 + n_1, q < 0.50 \end{cases} \tag{12}$$

$$m_2 = E \times (GH_i(t) - GH_r(t)) \tag{13}$$

$$n_2 = F \times (GH_i(t) - GH_m(t)) \tag{14}$$

$$E = (2 \times r_4 - 1) * a \tag{15}$$

$$F = (2 \times r_5 - 1) * b \tag{16}$$

where $MGH_i(t + 1)$: new location of gannet; r_4 and r_5 : random numbers; m_1 : a random number between $-a$ and a ; n_1 : a random number between $-b$ and b ; $GH_i(t)$: the location of the i th gannet; $GH_m(t)$: the average location; and $GH_r(t)$: the random location of gannet. Based on two scenarios, gannets pursue fish. The first scenario assumes that the gannet has enough energy to chase fish. However, the second scenario assumes that the gannet lacks the energy to chase fish [28]. In order to determine the energy level of gannets, we define a parameter called capture ability. Equation (17) is used to define the capture ability parameter. This equation consists of two controller parameters. Equations (19) and (20) are used to define values of controller parameters.

$$capture(ability) = \frac{1}{R * \chi_2} \tag{17}$$

$$\chi_2 = 1 + \frac{IT}{IT_{MAX}} \tag{18}$$

$$R = \frac{M * VEL^2}{K} \tag{19}$$

$$K = 0.20 + (2 - 0.2) \times r_6 \tag{20}$$

where r_6 : random number, M = mass of gannet (2.5), VEL : velocity (1.5 m/s), and r_6 : random number. When the gannet’s energy level exceeds the threshold, the position of the gannet is updated, and the fish is captured. Otherwise, the gannet randomly moves [28].

$$MGH_i = \begin{cases} \chi \times delat(GH_i - GH_{best}(t)) + GH_i(t), (capture)(ability) > v \\ GH_{best} - (GH_i(t) - GH_{best}(t)) \times F \times \chi, (capture)(ability) < v \end{cases} \tag{21}$$

$$delta = capture(ability) * |GH_i(t) - GH_{best}| \tag{22}$$

$$F = Levy(Dim) \tag{23}$$

where v : constant value ($v = 0.20$), $GH_{best}(t)$: the best location of gannet, and $Levy(Dim)$: levy flight function.

$$Levy(Dim) = 0.01 \times \frac{\mu \times \sigma}{|\kappa|^{\frac{1}{\tau}}} \tag{24}$$

$$\sigma = \left(\frac{\Gamma(1 + \psi) \times \sin\left(\frac{\pi\psi}{2}\right)}{\Gamma\left(\frac{1+\psi}{2}\right) \times \psi \times 2^{\left(\frac{\psi-1}{2}\right)}} \right)^{\frac{1}{\psi}} \tag{25}$$

where μ and σ : random values and ψ : constant value. A comparison is made between GOA and several algorithms to evaluate its capabilities in selecting inputs and adjusting model parameters. The following algorithms are selected as they can provide accurate solutions.

2.4.2. Bat Algorithm (BA)

The bat optimization algorithm was inspired by bat life. The echolocation ability of bats helps them locate prey [29]. The bat makes loud noises and receives echoes from its

surroundings to find prey [30]. The optimization process begins with the installation of frequency. Bats change their velocity at each iteration.

$$fr_i = fr_{\min} + (fr_{\max} - fr_{\min})\varepsilon \quad (26)$$

$$VE_i^t = VE_i^{t-1} + (X_i^{t-1} - X_i^*)fr_i \quad (27)$$

where ε : a random value; fr_i : the i th frequency; fr_{\min} : minimum frequency; fr_{\max} : maximum frequency; VE_i^t : the i th velocity at iteration t ; VE_i^{t-1} : the i th velocity at iteration $t - 1$; X_i^{t-1} : the position of the i th agent at iteration $t - 1$; and X_i^* : the best location of the bat. The bats perform a local search based on loudness:

$$X_i^{new} = X_i^{old} + \zeta A^t \quad (28)$$

where X_i^{new} : the new location of bat; X_i^{old} : the old location of bat; A^t : loudness; and ζ : a random number. Finally, we update loudness and rate of pulse emission (RAP) at each iteration:

$$A_i^t = \zeta A_i^{t-1} \quad (29)$$

$$RAP_i^t = RAP_i^0 + (1 - \exp(-\gamma t)) \quad (30)$$

where A_i^{t-1} : the i th loudness at iteration $t - 1$; RAP_i^0 : the previous value of RAP, and RAP_i^t : the new value of RAP.

2.4.3. Particle Swarm Optimization (PSO)

The PSO was inspired by the behavior particles. The PSO is selected for optimization and input selection due to its simple structure and high accuracy [31]. The velocity of the particles is updated using Equation (34) [32]:

$$VE_i^{t+1} = \omega \times VE_i^t + r_1 \times \alpha_1 \times (LO_{pbest} - LO_i^t) + r_2 \times \alpha_2 \times (L_{Gbest} - L_i^t) \quad (31)$$

$$LO_i^{t+1} = LO_i^t + VE_i^t \quad (32)$$

where VE_i^{t+1} : the particle velocity i ; particle number, $t + 1$: iteration $t + 1$; LO_{pbest} : the local best solution; L_{Gbest} : the global best solution; ω : weight coefficient; α_1 and α_2 : acceleration coefficients; LO_i^{t+1} : the best location; and r_1 and r_2 : random parameters. The optimization process begins with the installation of an initial population of particles. For updating the particle's velocity and position, Equations (31) and (32) are used. Until convergence is reached, the optimization process continues.

2.4.4. Genetic Algorithm (GA)

GA is one of the most popular algorithms for solving optimization problems. GA consists of a group of chromosomes [33]. Three operators, namely selection, mutation, and crossover, are used to maintain population diversity. The selection operator is used to select the best solution. Crossover plays a key role in the evolution of solutions. A crossover operator combines genetic information to create new offspring [34]. The mutation operator randomly alters genes to maintain genetic diversity in a population.

2.4.5. Binary Version of Optimization Algorithms

Binary algorithms select inputs. Conversion of a continuous algorithm to a binary version is necessary for feature selection. Transfer functions convert continuous values to binary values [35]. The S-shaped function is used as a transfer function. Previous articles reported that the S-shaped transfer function gave the most accurate results [36].

$$S(GH_{m,j}^t) = \frac{1}{1 + e^{-x_{m,j}^t}} \quad (33)$$

$$GH_{m,j}^t = \begin{bmatrix} 1 \leftarrow rand < S(GH_{m,j}^t) \\ 0 \leftarrow rand \geq S(GH_{m,j}^t) \end{bmatrix} \quad (34)$$

where *rand*: random number, $S(GH_{m,j}^t)$: the transfer function value, and $GH_{m,j}^t$: the *j*th solution in the *m*th position.

2.5. Hybrid CLM Model for Predicting GWL

Our study uses the CLM model to predict GWLs. We follow the following steps to create the hybrid CLM model.

- The parameters of the CNN model and the names of the inputs are initialized as the initial population of the binary GOA. In the modeling process, it is important to determine the training and testing data. K-fold cross-validation (KCV) is one of the suitable methods for partitioning input data into training and testing datasets [37]. The KCV method evaluates model accuracy by dividing data into K folds. Training data are divided into K-1 folds. Finally, the model is run K times to calculate the average error function. The average error function value is used to evaluate model performance and determine the optimal number of folds [38].
- Each possible input scenario is encoded as a binary string. Each bit represents the presence (1) or absence (0) of a specific input variable. An initial population of potential solutions (binary strings) is generated. These strings represent various combinations of input variables. The binary string 11,111 indicates that all inputs are considered during modeling. The algorithm can identify the most influential input combinations based on the defined fitness function by exploring various combinations and evaluating their performance.
- The CNN model is executed by using training data.
- CNN is run in the testing phase if termination criteria are met; otherwise, CNN is connected to GOA. CNN parameters and input names are updated using optimization algorithms.
- LSTM models use CNN outputs as inputs. The LSTM model is run using training data. The LSTM parameters are defined as the initial population of optimizers. The LSTM model is run at the testing level if the termination criteria are met; otherwise, the GOA is connected to the LSTM mode.
- MLR uses the outputs of the LSTM model as inputs. The LSTM model extracts temporal features from time series data. Temporal features can capture relationships between different data points. Then, the extracted features are sent to the MLR model to predict GWLs. The MLR model produces final outputs.

Our study uses the RMSE as a fitness function. By finding input combinations with the smallest prediction errors, RMSE is minimized. The GWL prediction model is trained and tested using the appropriate subset of input variables (denoted by “1” in the binary string). The model’s predictions are compared with the actual GWL values. The RMSE value is then calculated to determine the quality of the selected input combination. If a solution violates constraints (upper and lower values of decision variables), we should correct the decision variables to achieve feasible solutions. When an infeasible solution is identified, boundary correction methods are applied to modify the solution. For example, if a decision variable x_i violates its upper or lower bounds, it can be updated using Equation (35). Figure 1 shows the values of decision variables and constraints.

$$x_i = \text{Min}(\max(x_i, \text{Lower}(\text{Bound})), \text{Upper}(\text{Bound})) \quad (35)$$

If the current value of x_i is below the lower bound, Equation (35) uses the maximum value between x_i and the lower bound to correct it. If the current value of x_i exceeds

the upper bound, Equation (35) uses the minimum between x_i and the upper bound to correct it.

We benchmark the CLM model against the LSTM-MLR (LSM), CNN-MLR (CNM), CNN-LSTM (CNL), LSTM, CNN, and MLR models. In order to create the LSM, LTC, and CNM models, the input data are fed into the first model. Then, the outputs of the first model are fed into the second model.

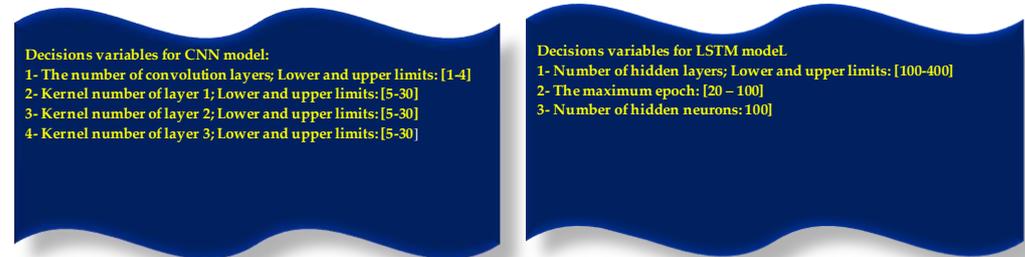


Figure 1. Values of decision variables and constraints.

2.6. Quantitation of Uncertainty Values

The input parameter is one of the most important sources of uncertainty. The model parameter is another source of uncertainty [39]. Therefore, it is important to quantify output uncertainty based on input and model parameters. The normal distribution is used as a prior distribution of input parameters. Normal distributions cannot be used as prior distributions for model parameters, because they may not have any physical meaning [39]. We study the variability of parameters during the training process to calculate the prior distribution of model parameters. The Monte Carlo simulation method is applied to regenerate sample parameters, including model and input parameters. Then, the Nash–Sutcliffe model efficiency (NSE) coefficient is applied to estimate the likelihood function value. If the likelihood value of a parameter is below a threshold, the parameter is removed from the calculation cycle. Finally, the posterior distribution of the parameter is calculated based on the prior distribution and the likelihood function.

2.7. Determination of Contribution of Input Parameters and Models to Output Uncertainty

In order to study the effect of input parameters and models on output uncertainty, it is necessary to decompose output uncertainty into parameter uncertainty and input uncertainty. The analysis of variance (ANOVA) method can be used to decompose output uncertainty [40]. The ANOVA method decomposes the total sum of squared errors (SSE) to determine the percentage contribution of each uncertainty factor [40]. Equations (34)–(40) explain the mathematical model of ANOVA.

$$SSE = \sum_{i=1}^N \sum_{j=1}^M (GWL^{ij} - GWL^{0,0})^2 \tag{36}$$

$$SSE = SS(IN) + SS(PA) + SS(IN_PA) \tag{37}$$

$$SS(IN) = M \cdot \sum_{i=1}^N (GWL^{i,0} - GWL^{0,0})^2 \tag{38}$$

$$SS(PA) = N \cdot \sum_{j=1}^M (GWL^{0,j} - GWL^{0,0})^2 \tag{39}$$

$$SS(IN_PA) = \sum_{h=1}^N \sum_{j=1}^M (GWL^{hj} - GWL^{i,0} - GWL^{0,j} + GWL^{0,0})^2 \tag{40}$$

where GWL^{ij} : predicted GWL based on i th input and j th model parameter; $GWL^{i,0}$: predicted GWL based on i th input; $GWL^{0,j}$: predicted GWL based on j th parameter; $GWL^{0,0}$:

the mean monthly GWL; N : the number of input parameters; M : the number of model parameters; $SS(IN_PA)$: the SSE based on the interaction between input parameter and model parameter; $SS(IN)$: the SSE based on the input parameter; and $SS(PA)$: the SSE based on the model parameter.

Finally, we use Equations (40)–(42) to determine the contribution of the uncertainty resource to the total uncertainty:

$$\eta_{IN}^2 = \frac{SS(IN)}{SSE} \quad (41)$$

$$\eta_{PA}^2 = \frac{SS(PA)}{SSE} \quad (42)$$

$$\eta_{IN_PA}^2 = \frac{SS(IN - PA)}{SSE} \quad (43)$$

where η_{IN}^2 : the percentage contribution of input parameter to the output uncertainty; η_{PA}^2 : the percentage contribution of model parameter to the output uncertainty; and $\eta_{IN_PA}^2$: the percentage contribution of the interaction between inputs and model parameters to the output uncertainty.

2.8. Trend Analysis

Water resources management requires trend detection of groundwater levels (GWL). GWL trend detection helps determine groundwater sustainability and availability in basins. Our study uses the Innovative-Şen trend (İŞT) method for trend analysis of GWLs [41]. Time-series data are split into two equal sets. The data of the first and second half of a time series are plotted on the x -axis and y -axis. The slope is calculated using Equation (44):

$$S = \frac{2(X_2 - X_1)}{N} \quad (44)$$

where S : slope; X_2 : the average value of the second subset; X_1 : the average value of the first subset; and N : the number of data. If $|S|$ is higher than $|S_{cr}|$ (critical slope), the trend is significant.

$$S_{cr} = 1.96 \times \sigma \quad (45)$$

σ : Standard deviation.

3. Case Study

3.1. Details of Case Study

Yazd-Ardekan, located in the desert, has a dry and hot climate. Water scarcity is a serious challenge in the study area due to the dry climate and excessive groundwater consumption. There has been a reduction in aquifer storage by 65 MCM. To predict the groundwater level in the study, the GWL of all piezometers is estimated. Thiessen polygons are defined by the geometric scope of each piezometer. Then, the area of each polygon is multiplied by the groundwater level of the corresponding piezometer. This process is repeated for all piezometers. The obtained outputs are summed together, and their average value represents the groundwater level in the area. Figure 2a,b show the study area and time series data. Table 2 lists the characteristics of data points. The lagged GWL data and climate parameters are used to predict one-month-ahead GWLs. Climate parameters include wind speed (WSP), evaporation (EVA), relative humidity (REH), rainfall, and temperature (TEM). There are 1194 pumping wells in this basin. Overexploitation of the aquifer reduces groundwater storage by 65.93 MCM.

Since the study area experiences hot and dry summers, water supply is one of the real challenges for decision-makers. The months of July, June, and May have the lowest rainfall amounts. Furthermore, these months have the highest evaporation rates observed. The data of 71 piezometers were used to predict GWLs. The data were collected from 1995 to 2010.

Our study used different inputs, such as lagged climate and GWL data, to predict GWLs. Data augmentation involves creating modified or synthetic versions of existing data to increase the dataset's size, diversity, and richness. Feature augmentation enhances the richness of the input data and helps the model learn from the temporal dependencies and trends of the historical data. A lagged data set consists of features that represent historical GWL measurements over different periods of time. Adding lagged data to the dataset can enrich the information available. This technique can enhance the model's understanding of temporal dependencies and patterns of GWL time series data. While feature augmentation using lagged data is valuable for predicting GWLs, it does not create new data samples or artificially augment the dataset. The next studies can use augmentation methods such as window slicing, time wrapping, and data decomposition to improve model performance.

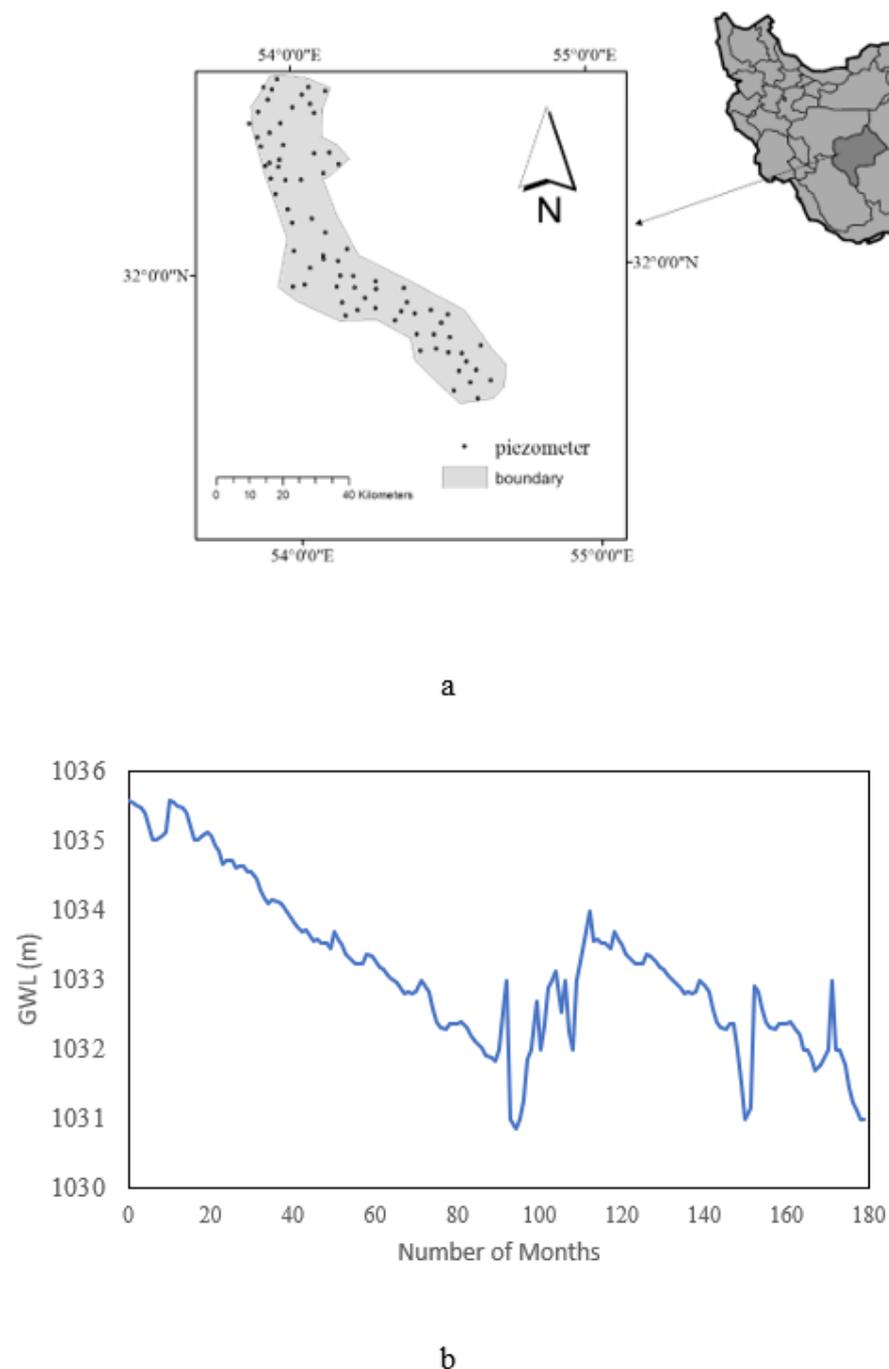


Figure 2. (a): Location of Plain and (b): GWL time series during period 1995–2010.

Table 2. Characteristics of monthly data.

| Parameter | Maximum | Average | Minimum |
|------------|---------|----------|---------|
| (EVA) (mm) | 51 | 31 | 14 |
| (RELH)% | 75 | 56 | 44 |
| (RA) (mm) | 20 | 8 | 0 |
| WIS (m/s) | 14.01 | 11.59 | 2.65 |
| (TEM) °C | 41 | 25 | 7.4 |
| GWL (m) | 1035.52 | 1033.281 | 1031.23 |

Our study uses Equations (46)–(53) to explore the potential of models for predicting GWLs:

- Root mean square error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (GW_{es} - GW_{ob})^2}{N}} \tag{46}$$

- Mean absolute error (MAE)

$$MAE = \sqrt{\frac{\sum_{i=1}^N (GW_{es} - GW_{ob})^2}{N}} \tag{47}$$

- Nash Sutcliffe efficiency (NSE)

$$NSE = 1 - \frac{\sum_{i=1}^N (GW_{ob} - GW_{es})}{\sum_{i=1}^N (GW_{ob} - \bar{GW})} \tag{48}$$

- Percent bias (PBIAS)

$$PBIAS = \frac{\sum_{i=1}^N (GW_{ob} - GW_{es})}{\sum_{i=1}^N GW} \tag{49}$$

- Reliability index (REI)

$$RE = \frac{\sum_{i=1}^N J_i}{N} \times 100 \tag{50}$$

$$J_i = \left[\begin{array}{l} 1 \leftarrow \text{if}(RE_i \leq \delta) \\ 0 \leftarrow \text{else} \end{array} \right] \tag{51}$$

$$RAE_i = \frac{|GW_{es} - GW_{ob}|}{GW_{ob}} \times 100 \tag{52}$$

- Resiliency index (RES)

$$RES = \left[\begin{array}{l} 100\% \leftarrow \text{if}(\text{Reliability} = 100\%) \\ \frac{\sum_{i=1}^{N-1} R_i}{N - \sum_{i=1}^N J_i} \times 100 \end{array} \right] \tag{53}$$

$$R_i = \left[\begin{array}{l} 1, \text{if}(RAE_i > \delta(\text{and})RAE_{i+1} \leq \delta) \\ 0 \leftarrow \text{else} \end{array} \right] \tag{54}$$

where GW_{es} : estimated GWL; GW_{ob} : observed data; N : the number of data; and δ : a constant value (δ based on the Chinese standard = 0.20). Equations (55)–(57) are applied to quantify uncertainty values.

- Width interval (WI)

$$WI = \sum_{i=1}^n \frac{U_i - L_i}{nR} \tag{55}$$

- Prediction interval coverage probability (PICP)

$$PICP = \frac{1}{n} \sum_{i=1}^n \kappa_i \tag{56}$$

$$\kappa_i = \begin{bmatrix} 1 \leftarrow y_i \in [L_i, U_i] \\ 0 \leftarrow y_i \notin [L_i, U_i] \end{bmatrix} \tag{57}$$

where L_i and U_i : upper and lower values of variables, y_i : observed data, and R : the difference between maximum and minimum values.

3.2. Error Function for Evaluating the Performance of Optimization Algorithms in Choosing Inputs

There is a good balance between the exploration and exploitation capabilities of an ideal algorithm. Equations (58) and (59) are used to compute the dimension-wise diversity. Equations (60) and (61) are used to compute the percentage of exploration and exploitation:

$$div_j = \frac{1}{N} \sum_{i=1}^N medain(s^j) - s_i^j \tag{58}$$

$$div^t = \frac{1}{D} \sum_{i=1}^D div_j \tag{59}$$

$$Exploration = \frac{div^t}{\max(div)} \tag{60}$$

$$exploitation = \frac{|div^t - \max(div)|}{\max(div)} \tag{61}$$

where s_i^j : the i th solution in the j th dimension, $medain(s^j)$: the median value of the j th dimension, D : the number of dimensions, and $\max(div)$: maximum diversity. The average selection size (ASS) index is used to evaluate the performance of each optimizer in selecting the best input scenario. The ideal algorithms select a minimum number of inputs that guarantee the highest accuracy for predicting GWL [42].

$$Average(selection(size)) = \frac{1}{M} \sum_{i=1}^M \frac{size(so^i)}{NU} \tag{62}$$

$$U_{95} = 1.96 \left(SD^2 + RMSE \right)^{\frac{1}{2}} \tag{63}$$

where M : The number of runs of optimization algorithms ($M = 1000$), NU : the number of total inputs, so^i : the size of chosen features at each iteration, and SD : standard deviation. The low values of U_{95} and ASS show the best algorithm for choosing inputs. We couple different algorithms with the CLM model to determine the optimal input scenarios and model parameters for predicting GWLs. The algorithms send a chosen input scenario to the CLM model to predict GWLs. Then, we compute the U_{95} of optimization algorithms to assess their performance in choosing inputs.

4. Results

4.1. Determination of Algorithm Parameters

Random parameters affect the performance of algorithms. The maximum number of iterations (MNOI) is one of the important random parameters of algorithms. In addition, population size is another important random parameter. Figure 3 shows the MAE values for different values of random parameters. The value of the error function was calculated for different population sizes and iteration numbers. The population size (PS) of 100 and

the MNOI of 200 yielded the lowest MAE value for GOA. The PS of 150 and the MNOI of 200 yielded the lowest MAE values for BA and PSO. The PS of 200 and the MNOI of 100 provided the lowest MAE values for GA. The ideal parameters provide the lowest error function. BA and PSO have different random parameters. The best values of random parameters can be determined through sensitivity analysis. Table 3a,b show the optimal values of the random parameters for BA and PSO. The maximum frequency of BA varied from three to nine. The maximum frequency of seven provided the lowest error function. Also, a maximum loudness of 0.70 gave the lowest error function values. Similarly, the optimal values of the PSO parameters are obtained.

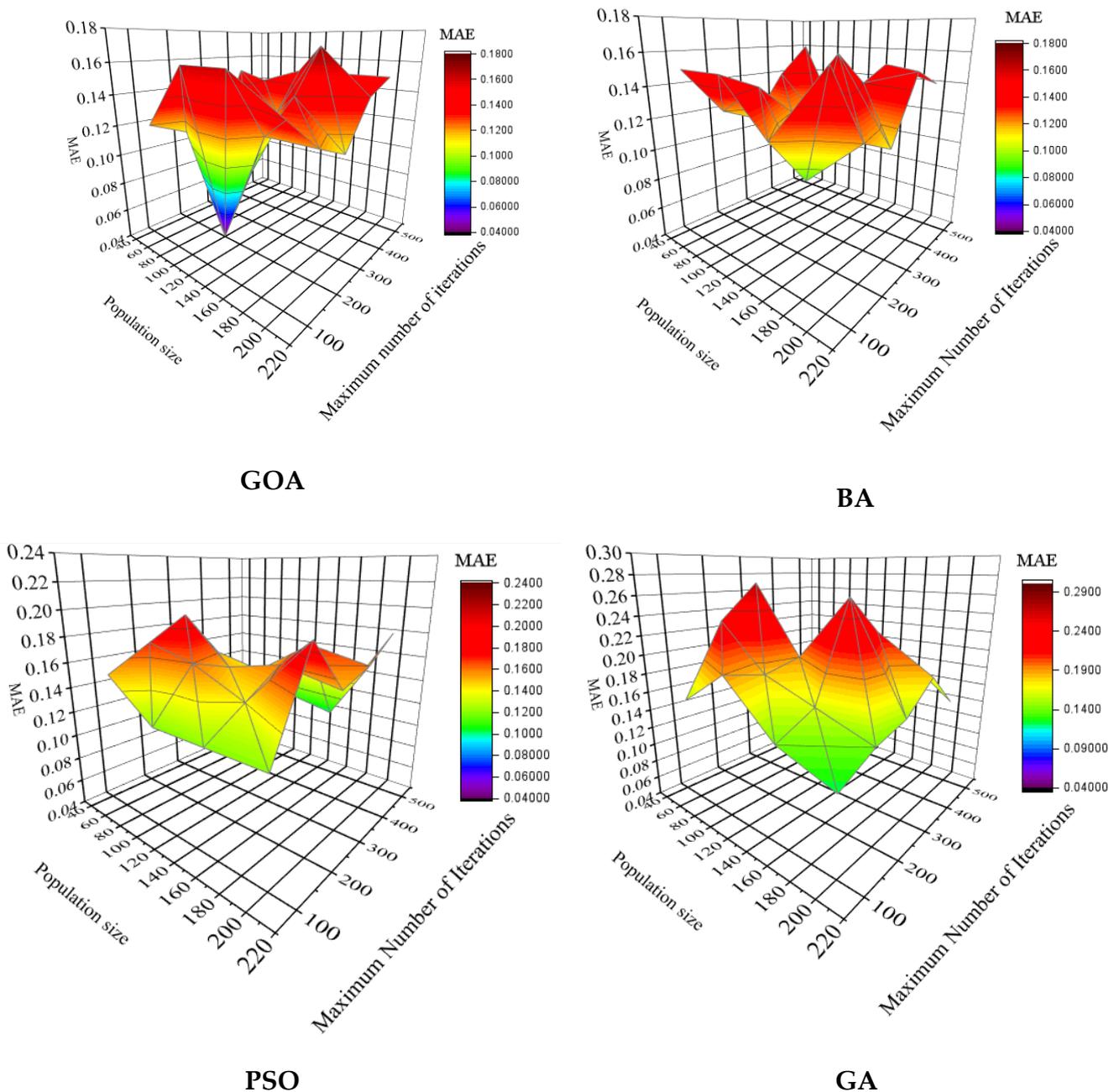


Figure 3. Determination of the population size and the maximum number of iterations of algorithms.

4.2. Determination of Number of Folds (K)

To run the models at training and testing levels, we need to determine the number of folds. Figure 4 displays the MAE values for various sizes of k. The iteration

number = 150 and $K = 10$ yielded the lowest values for the MLR and CNN models. The iteration number = 100 and $K = 10$ yielded the lowest values for the LSTM and CLM models. The iteration number = 100 and $K = 10$ provided an MAE value of 0.09 for the CNL model. $K = 10$ was chosen for all models to run at training and testing levels. The sizes that provide the lowest error function are ideal.

Table 3. (a): Determination of optimal values of BA parameters, (b): determination of optimal values of PSO parameters (c): U95 and ASS values of different algorithms, and (d): optimal values of model parameters.

| (a) | | | | | |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------------|------------------|-----------|
| Maximum frequency | MAE value | Minimum frequency | MAE value | Maximum Loudness | MAE value |
| 3 | 0.14 | 0 | 0.15 | 0.5 | 0.15 |
| 5 | 0.12 | 1 | 0.12 | 0.6 | 0.12 |
| 7 | 0.09 | 2 | 0.10 | 0.7 | 0.09 |
| 9 | 0.15 | 3 | 0.14 | 0.8 | 0.12 |
| (b) | | | | | |
| α_1 | MAE value | α_2 | MAE value | | |
| 1.6 | 0.17 | 1.6 | 0.18 | | |
| 1.8 | 0.15 | 1.8 | 0.16 | | |
| 2.00 | 0.12 | 2.00 | 0.12 | | |
| 2.2 | 0.14 | 2.2 | 0.15 | | |
| (c) | | | | | |
| Algorithm | U95 | ASS | Standard deviation | | |
| GOA | 3.45 | 0.04 | 0.32 | | |
| BA | 5.75 | 0.07 | 0.67 | | |
| PSO | 8.25 | 0.08 | 0.78 | | |
| GA | 9.12 | 0.12 | 0.82 | | |
| (d) | | | | | |
| Model | Optimal values of model parameters | | | | |
| CNN | The number of convolution layers: 3, Kernel number of layer 1: 20; Kernel number of layer 2: 20, and Kernel number of layer 3: 15, batch size:30 | | | | |
| LSTM | Number of hidden layers:200, The maximum epoch:50, Number of hidden neurons: 100 | | | | |

4.3. Selection of the Best Optimization Algorithm for Choosing Inputs

Table 3c displays the U95 and ASS values of various algorithms. The U95 values of GHO, BA, PSO, and GA were 3.45, 5.75, 8.25, and 9.12, respectively. The ASS values of GOA, BA, PSO, and GA were 0.04, 0.07, 0.08, and 0.12, respectively. The GAO had the lowest ASS and U95 values. The GHO method could attain the highest accuracy. Additionally, the GOA method used fewer inputs to predict the GWL. Figure 5 evaluates the average exploration and exploitation percentages of different algorithms. A good optimization algorithm should strike a balance between exploration and exploitation capabilities. The algorithms were run 1000 times. The exploration and exploitation percentages of the GOA were 51% and 49%, respectively, while the exploration and exploitation percentages of the GA were 71% and 29%, respectively. The GOA achieved a good balance between exploration and exploitation abilities. Previous studies also showed that optimization algorithms could improve the accuracy of machine learning models [43,44]. Table 2d shows a list of model parameter values.

Our study used climate inputs and GWL data to predict the one-month-ahead GWL. The lag times of GWL data and climate data were variable from $t = 1$ to $t = 12$. Thus, there

are $72((\text{number of climate parameters} = 5) \times (\text{number of lag times} = 12)) + 12$ (lagged GWL data) input variables and $2^{72} - 1$ input combinations. The GOA chose the input combination of $\text{GWL}(t - 1)$, $\text{RELH}(t - 1)$, $\text{WIS}(t - 1)$, $\text{GWL}(t - 2)$, $\text{RA}(t - 1)$, and $\text{TEM}(t - 1)$ as the optimal input scenario for predicting GWLs. Previous studies also reported that lagged climate data and GWLs had high correlation values with GWLs [45]. Seasonal and monthly patterns of climate data can affect GWLs [46].

Feature selection is an important step in building accurate predictive models. It involves choosing the most relevant and informative input variables (features) from a large pool of candidates. Selecting the right features can significantly improve the model’s performance. GOA automates this process by determining the optimal input features for groundwater level prediction. GOA searches for the best combination of input features that minimizes prediction errors or maximizes the predictive performance metric (e.g., Nash–Sutcliffe efficiency coefficient or other relevant metrics used in the study). By selecting the most important features, GOA reduces the dimensionality of the input data. It is important because too many irrelevant or redundant features can lead to overfitting and computational inefficiency. GOA automates the selection of optimal input features. By selecting the most relevant features, GOA helps create more interpretable models. Features are used to improve model accuracy. The optimization process determines which features produce the best prediction of groundwater levels.

4.4. Evaluation of the Performance of Predictive Models in Predicting GWL

Figure 6a shows the NSE and PBIAS models for predicting GWLs. Heatmaps of NSE and PBIAS of different models were used to evaluate the performance of the models. The CLM improved the training NSE of the CNL, LSM, CNM, LSTM, CNN, and MLR models by 6.12%, 9.12%, 12%, 18%, 22%, and 30%, respectively. Testing NSE models of LSM and MLR were 0.94 and 0.68, respectively. The testing PBIASs of CLM, CNL, LSM, CNM, LSTM, CNN, and MLR were 7, 11, 14, 17, 21, 23, and 25, respectively. The study results showed that CLM improved the MLR model’s PBIAS. Figure 6b shows scatterplots of different models. The R2 values of the CLM, CNL, LSM, CNM, LSTM, CNN, and MLR were 0.9973, 0.9873, 0.9685, 0.9565, 0.9490, 0.9403, and 0.9351, respectively. It was observed that hybrid models improved LSTM, CNN, and MLR NSE. The CNM model improved the testing NSE values of CNN, LSTM, and MLR models by 4.08%, 14%, and 17%, respectively. The LSTM model enhanced the testing and training NSE of the CNN models by 5.2% and 7.6%, respectively. The CNN model improved the training and testing PBIAS values of the MLR model by 8.3% and 8.0%, respectively.

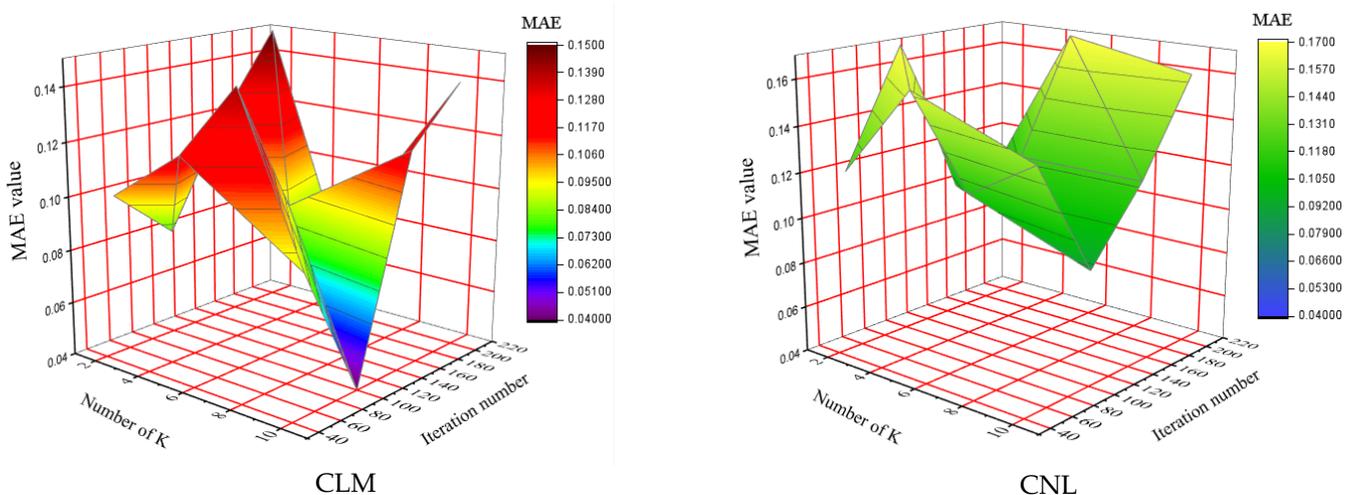


Figure 4. Cont.

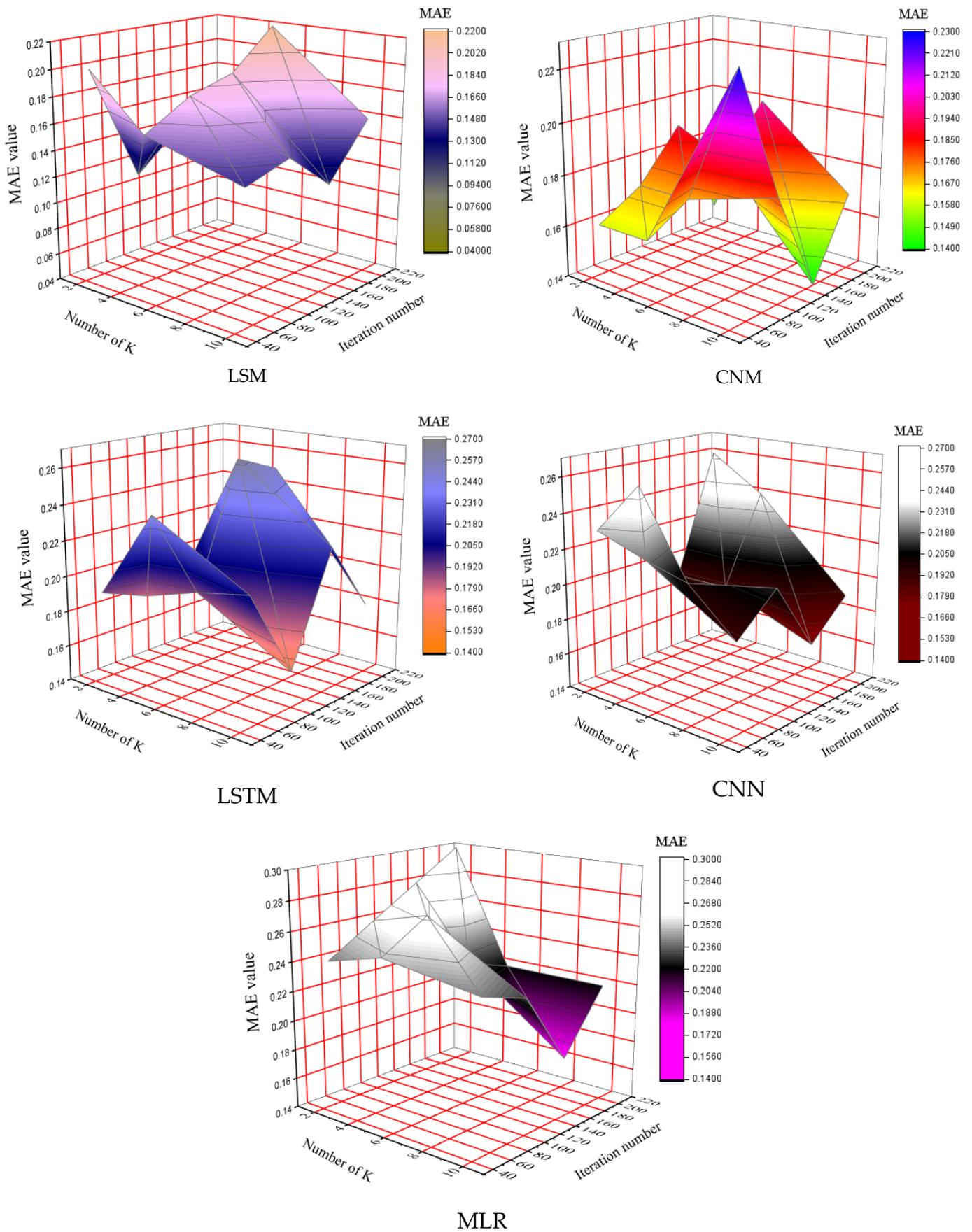


Figure 4. Determination of number of folds for the K-fold cross-validation (KCV) method.

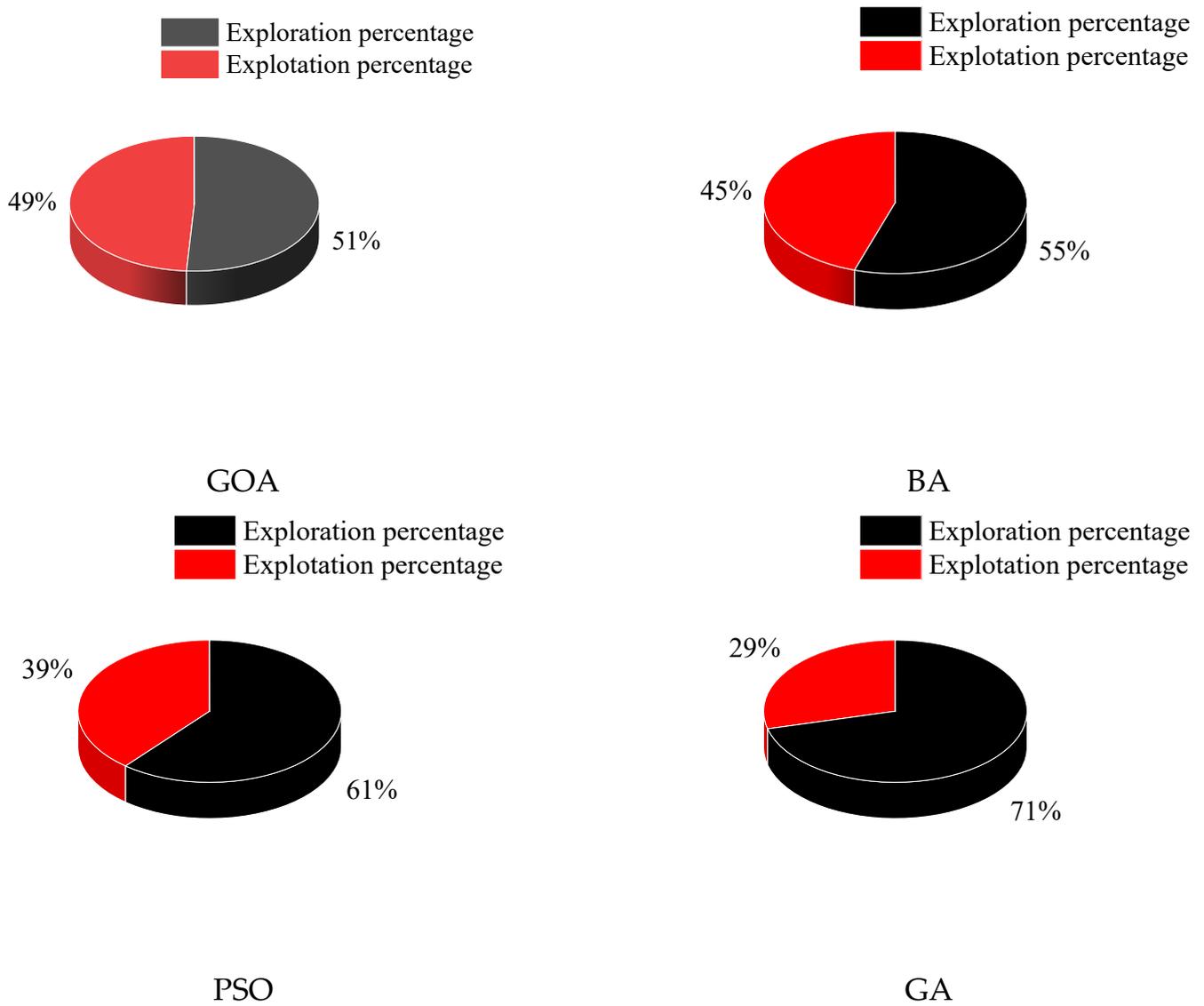
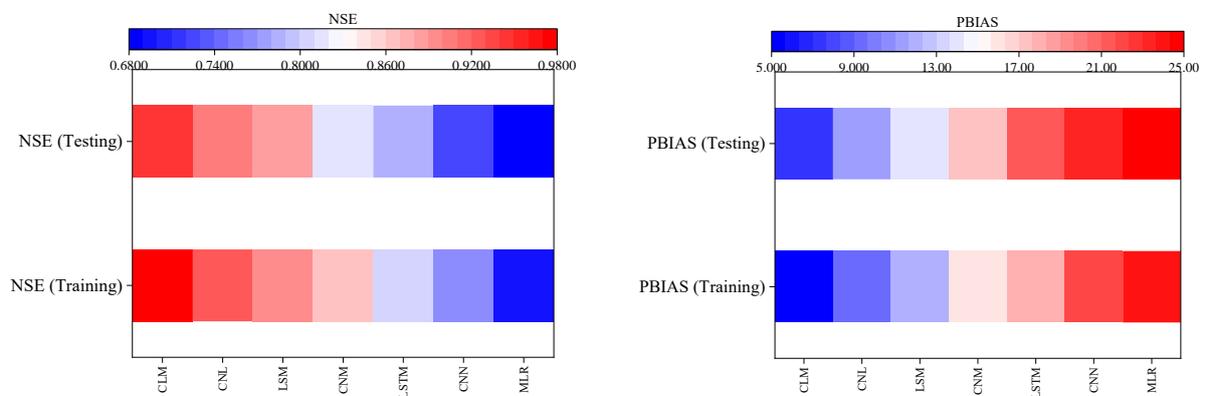


Figure 5. The average exploration and exploitation percentages of different algorithms.



(a)

Figure 6. Cont.

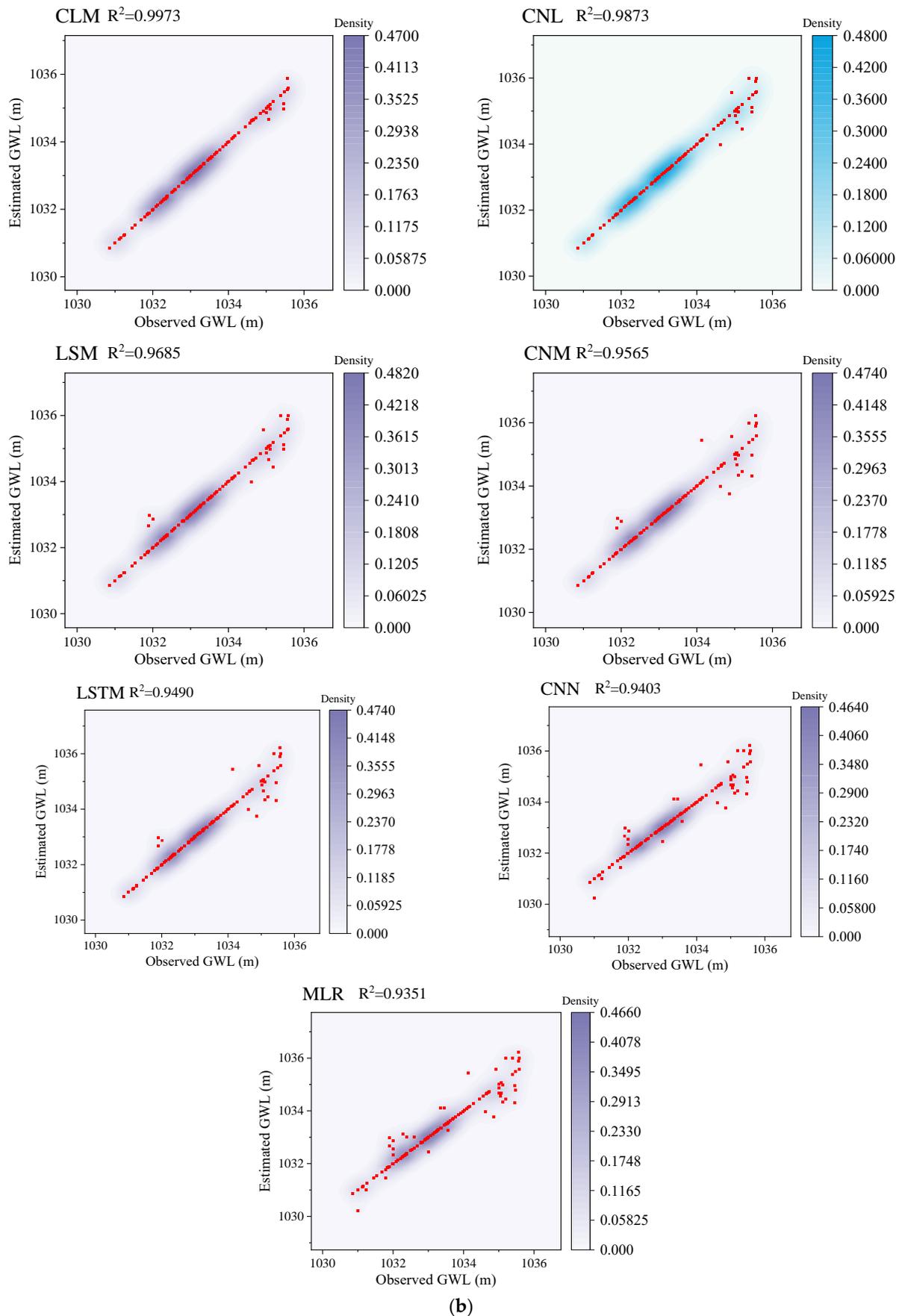
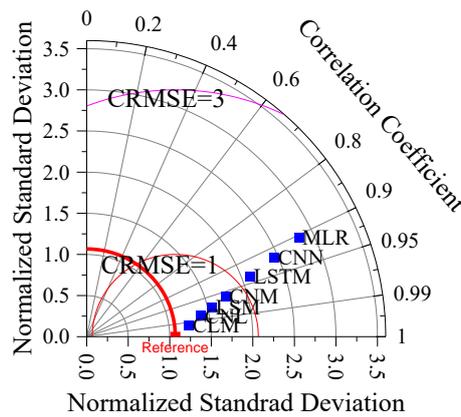


Figure 6. Cont.

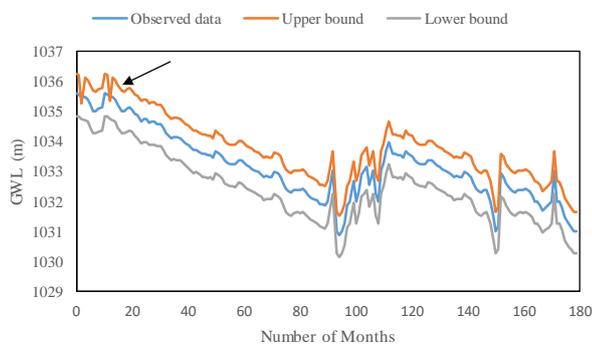


(c)

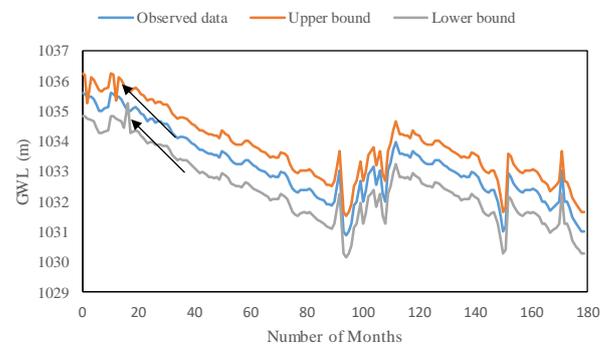
Figure 6. (a): Assessment of the efficiency of models based on NSE and PBIAS values, (b): scatterplots of models, and (c): Taylor diagram for assessment of the precision of models.

Taylor diagram is a graphical tool that assesses the accuracy of a data set or model compared to a reference data set. Taylor diagrams show each dataset or model as a point on a polar coordinate system. The distance between a point and the reference dataset is used to measure the standard deviation. The azimuth angle measures the correlation between the dataset/model and the reference point. The contour line shows different values of the centralized root mean square error (CRMSE). Figure 6c displays a Taylor diagram, which compares the accuracy of models. The CRMSEs of CLM, CNL, LSM, CNM, LSTM, CNN, and MLR were 0.19, 0.37, 0.52, 0.72, 1.07, 1.42, and 1.77, respectively. The correlation coefficients of CLM, CNL, LSM, CNM, LSTM, CNN, and MLR were 0.99, 0.98, 0.97, 0.95, 0.93, 0.91, and 0.90, respectively.

Figure 7 displays the uncertainty bounds of models based on two scenarios. The PICWs of the CLM model were 0.99 and 0.98 based on input and parameter uncertainties, respectively. The PICPs of the CNL, LSM, and CNM models were 0.97, 0.95, and 0.92 based on input uncertainty (IU). The PICPs of the CNL, LSM, and CNM models were 0.96, 0.92, and 0.91 based on parameter uncertainty (PU). Based on the IU, the WIs of the CLM, CNL, LSM, and CNM models were 0.03, 0.04, 0.07, and 0.12, respectively. Based on the PU, the WIs of the CLM, CNL, LSM, and CNM models were 0.05, 0.06, 0.09, and 0.14, respectively. The models provided a high uncertainty based on the input uncertainty. The CLM and MLR exhibited the largest and lowest uncertainty. The arrows indicate that the data points do not fall within the upper and lower bounds.

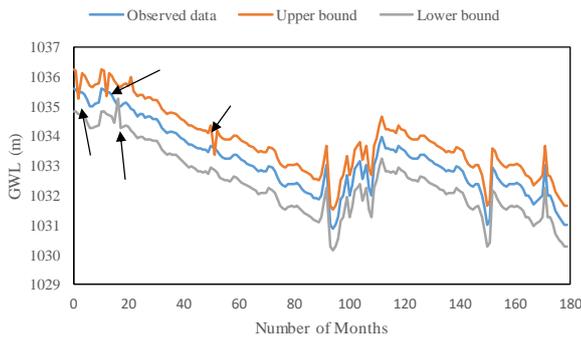


CLM (Input uncertainty, PCIP = 0.99, WI = 0.03)

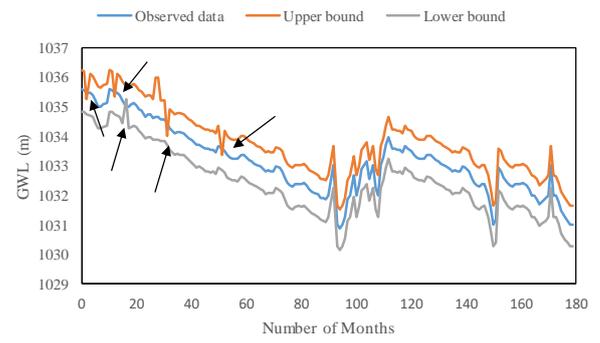


CLM (Parameter uncertainty, PICP = 0.05, WI = 0.05)

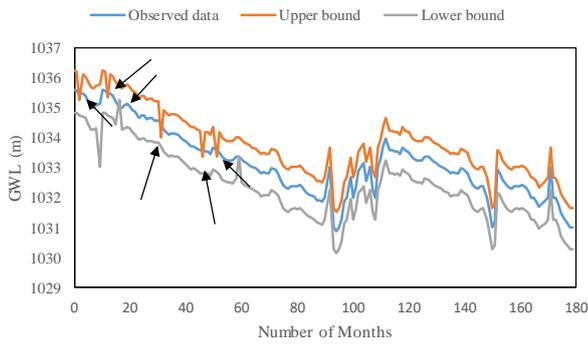
Figure 7. Cont.



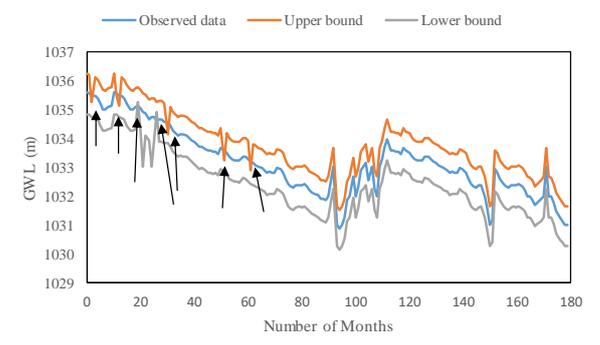
CNL (input uncertainty, PICP = 0.97, WI = 0.04)



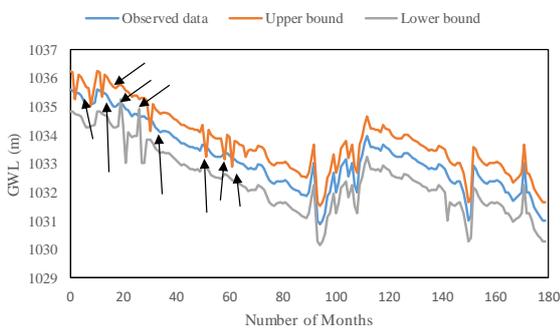
LSTM (parameter uncertainty, PICP = 0.92, WI = 0.09)



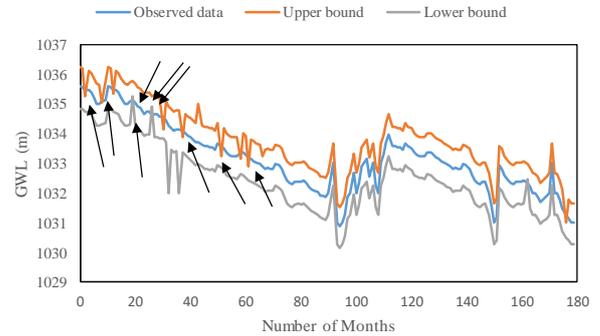
LSM (input uncertainty, PICP = 0.95, WI = 0.07)



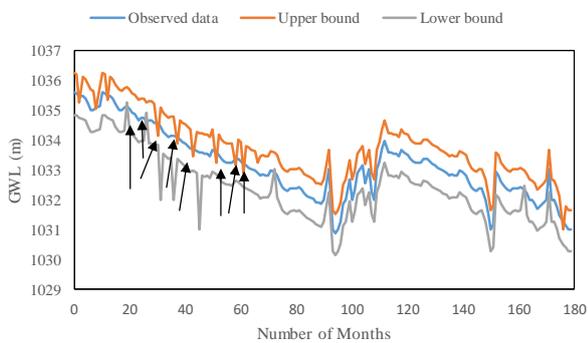
LSTM (parameter uncertainty, PICP = 0.92, WI = 0.09)



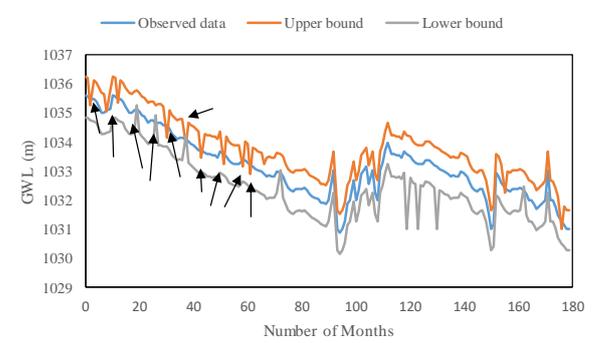
CNM (input uncertainty, PICP = 0.92 and WI = 0.12)



CNM (parameter uncertainty, PICP = 0.91 and WI = 0.14)



LSTM (Input uncertainty, PICP = 0.90, WI = 0.14)



LSTM (parameter uncertainty, PICP = 0.89, WI = 0.16)

Figure 7. Cont.

5. Discussion

5.1. Evaluation of the Accuracy of Models for Different Horizon Predictions

The CLM model was used for predicting the one-month ahead GWL. However, it is necessary to assess the performance of the new model over different time horizons. Figure 8a shows the REL index values for different time horizons. The CLM model was used for one-, three-, and six-month-ahead GWL predictions. The CLM model provided REL index values of 0.85–0.99, 0.76–0.96, and 0.69–0.96 for one-, three-, and six-month-ahead values, respectively. The CLM model provided RES index values of 0.87–0.96, 0.82–0.90, and 0.65–0.89 for one, three, and six months ahead. The results revealed that the accuracy of a predictive model decreased as the prediction horizon increased. The system can become more complex as the prediction horizon increases. Long-term trends and patterns can change over time because of various factors, such as climate change or human intervention. Therefore, all of these factors can affect model accuracy.

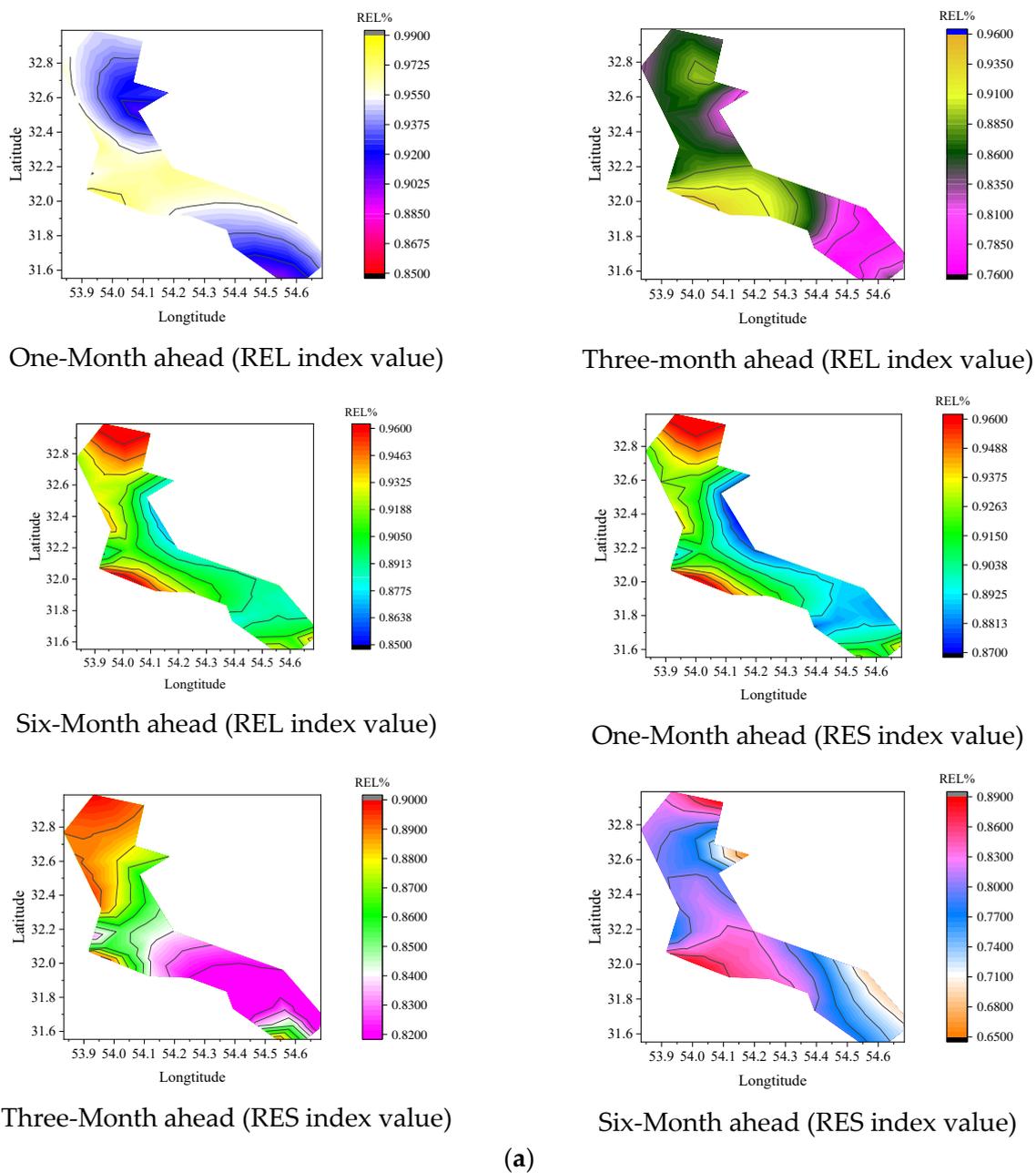


Figure 8. Cont.

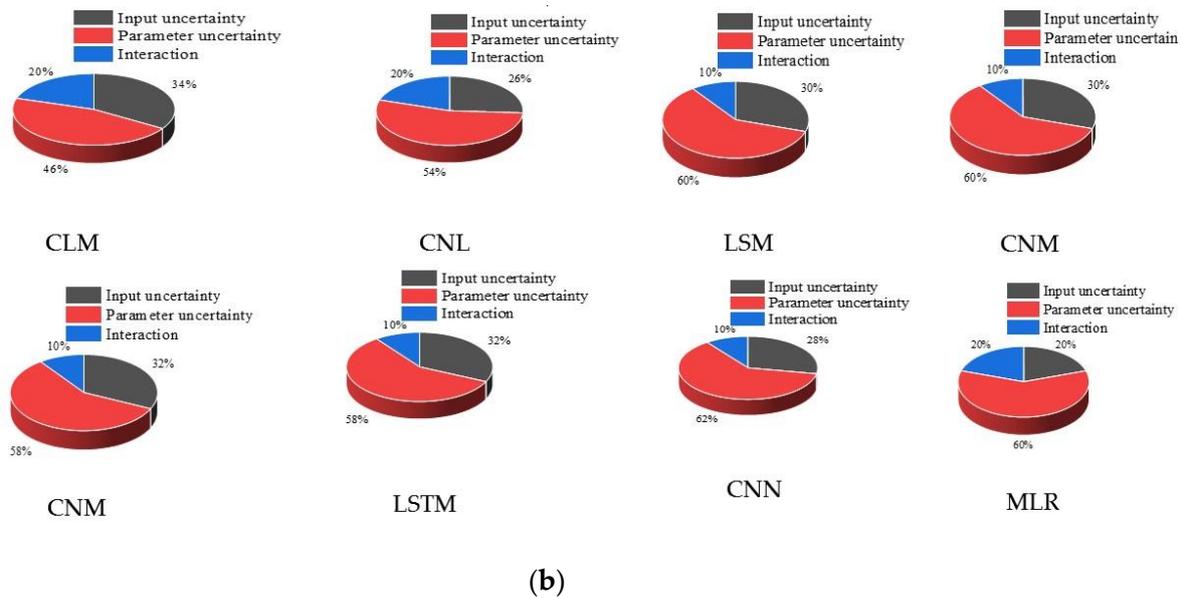


Figure 8. (a): The evaluation of the accuracy of models for predicting different horizons, and (b): the percentage contribution of uncertainty resources to output uncertainty.

5.2. The Contribution of Input and Parameter Uncertainty to the Output Uncertainty

The ANOVA method was utilized to determine the impact of inputs and model parameters on total uncertainty. Figure 8b illustrates the relative contribution of inputs and model parameters to total uncertainty. The CLM model had 46% parameter uncertainty, 34% interaction uncertainty, and 20% input uncertainty. MLR output uncertainty consisted of 60% parameter uncertainty, 20% parameter uncertainty, and 20% interaction uncertainty. The total uncertainty of the models was significantly influenced by parameter uncertainty.

5.3. Memory Usage Percentage of Models

The previous sections reported that the CLM model was more accurate than other models. However, it is necessary to evaluate the computational cost of the models. Figure 9 displays the percentage of memory usage of different models. The percentage of memory usage of the CNL, CNL, LSM, CNM, LSTM, CNN, and MLR was 42.9%, 39%, 37%, 35%, 30%, 27, 25%, and 20%, respectively. Although the CLM model requires more memory than other models, we can utilize advanced systems and memories to run it for predicting outputs.

5.4. Trend Defection

Figure 10 shows the S values of models. All months have negative trends. Summer months had lower S values than other months. June, July, and August had S values of $-0.04-(-0.12)$, $-0.03-(-0.12)$, and $-0.02-(-0.11)$, respectively. January Month had the lowest S values. S values of January month varied from -0.02 to (-0.04) . Winter months had higher S values than other months. The S_{Cr} shows critical S values. Climate and rainfall patterns can improve the GWL in the winter months.

The groundwater level showed a decreasing trend. However, different techniques can be used for groundwater recovery. Percolation ponds or basins can be constructed to allow surface water to infiltrate into the ground. Advanced monitoring technologies such as groundwater level sensors and remote sensing can be used to better understand groundwater dynamics.

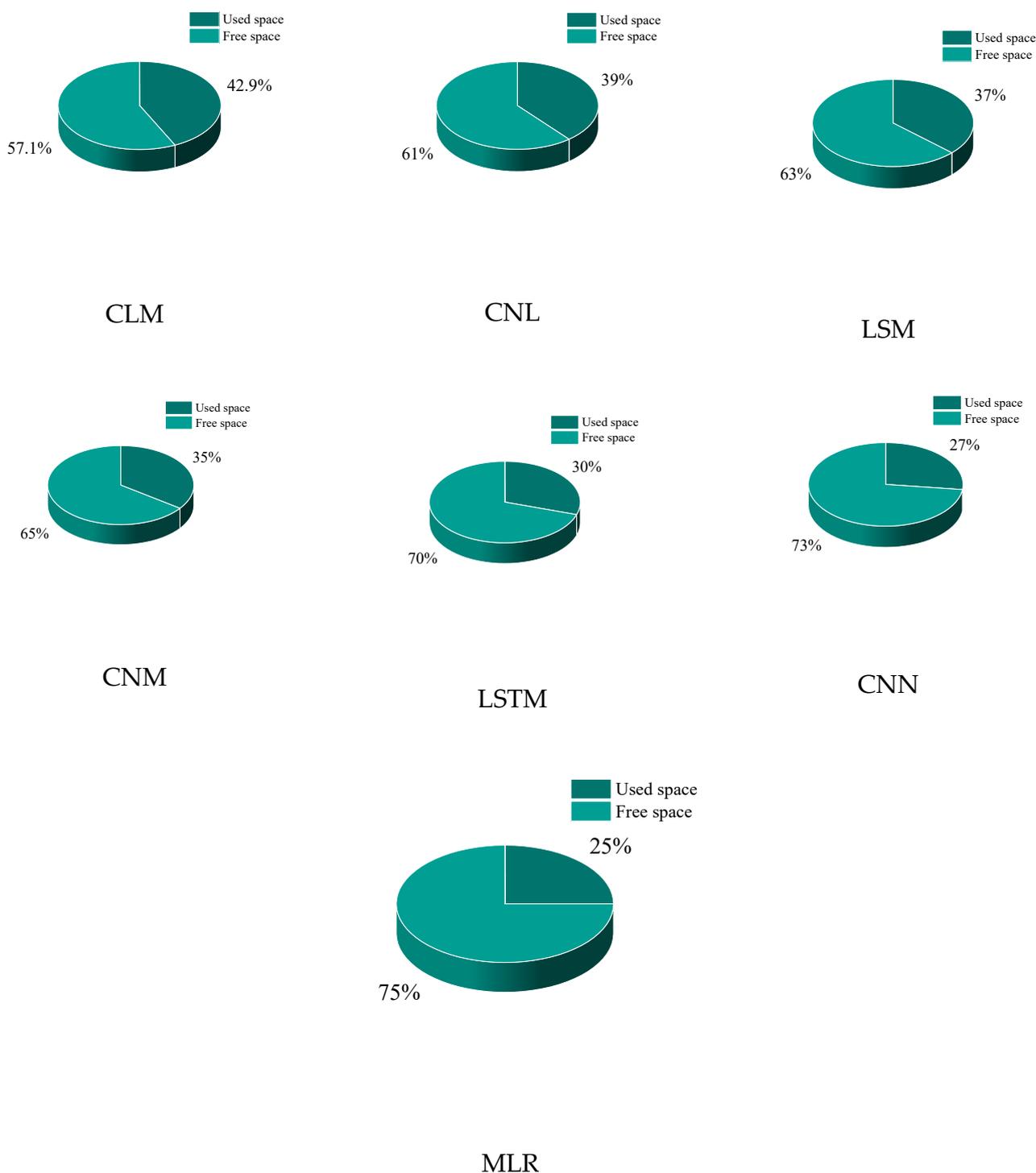


Figure 9. Memory usage of models.

Gathering data is one of the limitations of the study. However, different climate variables may not be available. Also, advanced computer systems are required to run models. A modeler may not have access to advanced systems. Also, adjusting model and algorithm parameters can be challenging as these parameters influence output accuracy. Although the suggested models can predict GWLs accurately, they may provide high uncertainty because of many parameters.

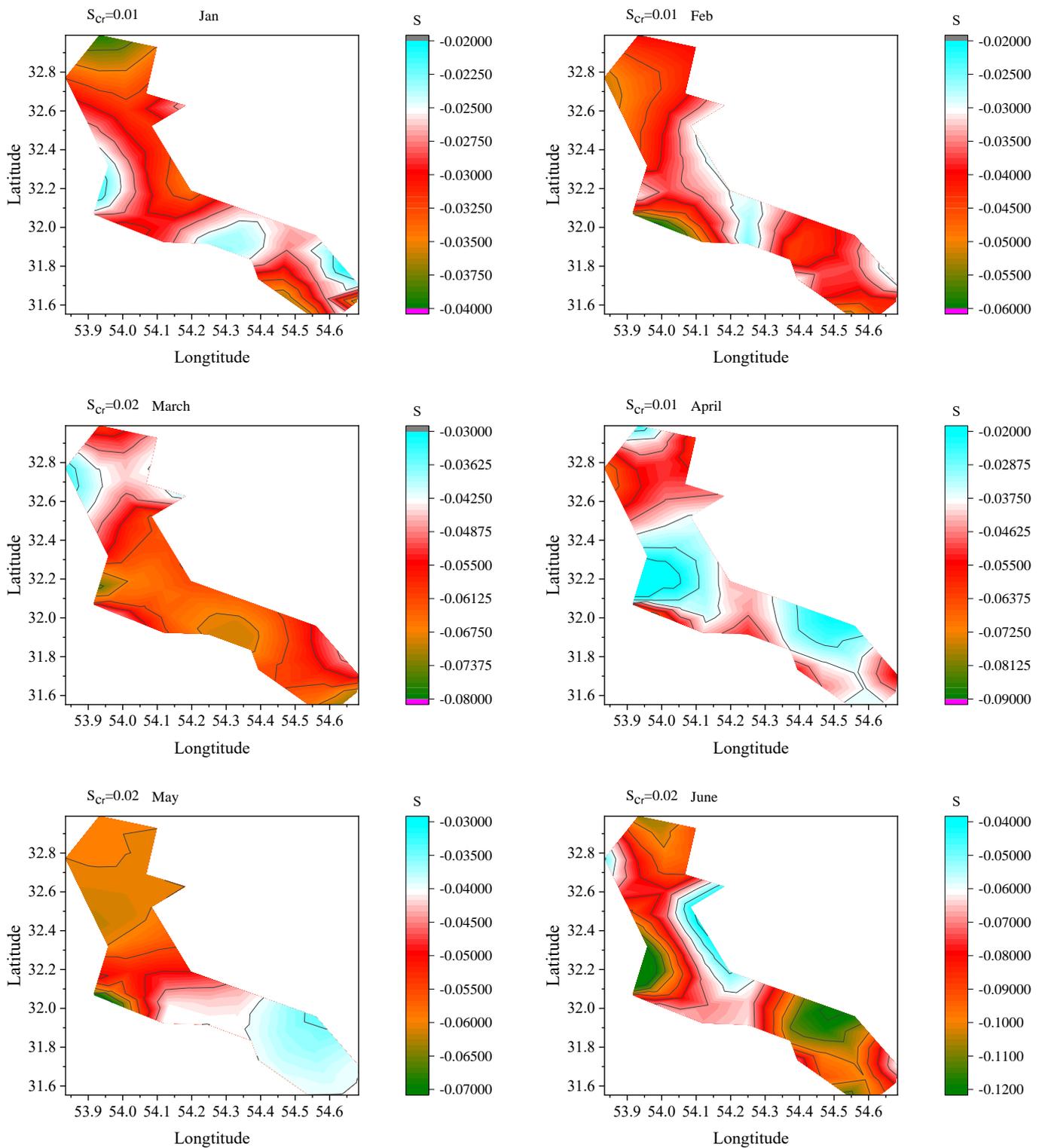


Figure 10. Cont.

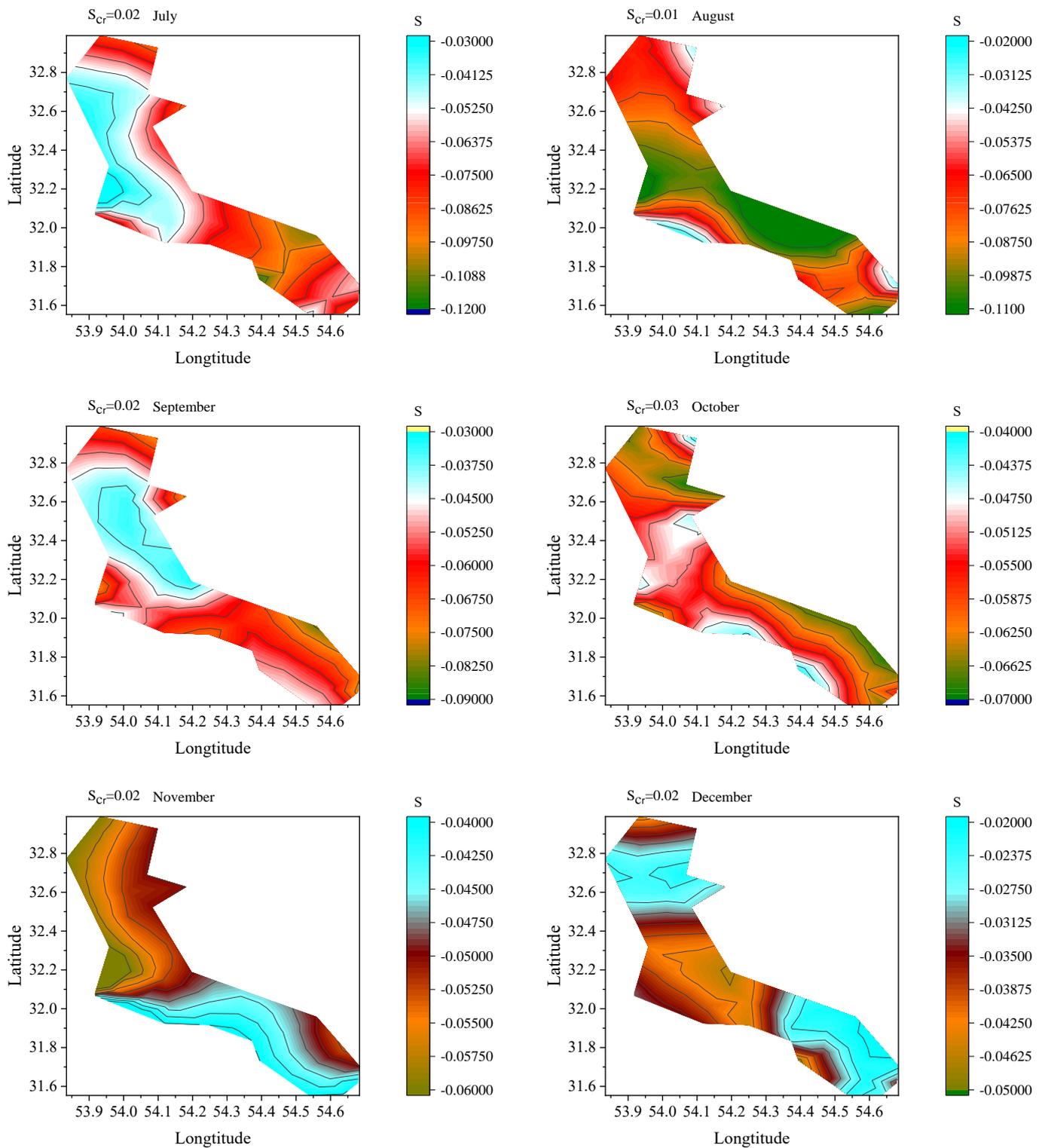


Figure 10. Trend of GWL in the basin.

6. Conclusions

Groundwater plays a key role in meeting irrigation and drinking water needs. Groundwater level prediction can be used for better water resource planning. Our study introduced a new model to predict one-month-ahead GWL. The CNN-LSTM-MLR (CLM) was developed to enhance the efficiency of the MLR model for predicting GWL. The CNN and LSTM models help the MLR model extract fetuses and predict GWL accurately. Different

optimizers were used to determine the best input scenarios. GOA outperformed other algorithms and successfully chose the inputs for predicting the GWL. The U95 values of GOA, BA, PSO, and GA were 3.45, 5.75, 8.25, and 9.12, respectively. Also, the ASS of the CLM was lower than that of the other algorithms. The study results indicated that CLM improved the training NSE of the CNL, LSTM, CNM, LSTM, CNN, and MLR models by 6.12%, 9.12%, 12%, 18%, 22%, and 30%, respectively. The study results indicated that CLM improved the training NSE of the CNL, LSTM, CNM, LSTM, CNN, and MLR models by 6.12%, 9.12%, 12%, 18%, 22%, and 30%, respectively. The uncertainty analysis indicated that the CLM model was the most reliable model for predicting GWL. The input parameters had less uncertainty than the model parameters. The total uncertainty of the CLM model consisted of 46% parameter uncertainty, 34% interaction uncertainty, and 20% input uncertainty. Long prediction horizons also decreased the accuracy of the CLM model. CLM can be used to predict GWL under different water stress conditions in the next studies. Also, the climate scenarios and models can be combined with the CLM model to predict GWL for feature periods. The next studies can also utilize the GOA algorithm as a robust feature selection method to determine optimal input scenarios and enhance the performance of deep learning models.

Author Contributions: Conceptualization, M.E. and F.B.B.; methodology, M.E.; software, M.E.; validation, M.E. and F.B.B.; formal analysis, M.E.; investigation, M.E.; resources, F.B.B.; data curation, writing—original draft preparation, M.E. and F.B.B.; writing—review and editing, M.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data are available based on reasonable requests.

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

References

1. Tao, H.; Hameed, M.M.; Marhoon, H.A.; Zounemat-Kermani, M.; Heddami, S.; Kim, S.; Sulaiman, S.O.; Tan, M.L.; Sa'adi, Z.; Mehr, A.D.; et al. Groundwater Level Prediction Using Machine Learning Models: A Comprehensive Review. *Neurocomputing* **2022**, *489*, 271–308. [[CrossRef](#)]
2. Di Nunno, F.; Granata, F. Groundwater Level Prediction in Apulia Region (Southern Italy) Using NARX Neural Network. *Environ. Res.* **2020**, *190*, 110062. [[CrossRef](#)]
3. Yadav, B.; Gupta, P.K.; Patidar, N.; Himanshu, S.K. Ensemble Modelling Framework for Groundwater Level Prediction in Urban Areas of India. *Sci. Total Environ.* **2020**, *712*, 135539. [[CrossRef](#)]
4. Khan, J.; Lee, E.; Balobaid, A.S.; Kim, K. A Comprehensive Review of Conventional, Machine Learning, and Deep Learning Models for Groundwater Level (GWL) Forecasting. *Appl. Sci.* **2023**, *13*, 2743. [[CrossRef](#)]
5. Saroughi, M.; Mirzania, E.; Vishwakarma, D.K.; Nivesh, S.; Panda, K.C.; Daneshvar, F.A. A Novel Hybrid Algorithms for Groundwater Level Prediction. *Iran. J. Sci. Technol. Trans. Civ. Eng.* **2023**, *47*, 3147–3164. [[CrossRef](#)]
6. Samani, S.; Vadiati, M.; Nejatjahromi, Z.; Etebari, B.; Kisi, O. Groundwater Level Response Identification by Hybrid Wavelet–Machine Learning Conjunction Models Using Meteorological Data. *Environ. Sci. Pollut. Res.* **2023**, *30*, 22863–22884. [[CrossRef](#)] [[PubMed](#)]
7. Mustafa, M.A.; Kadham, S.M.; Abbass, N.K.; Karupusamy, S.; Jasim, H.Y.; Alreda, B.A.; Al Mashhadani, Z.I.; Al-Hussein, W.R.A.; Ahmed, M.T. A Novel Fuzzy M-Transform Technique for Sustainable Ground Water Level Prediction. *Appl. Geomat.* **2023**, 1–7. [[CrossRef](#)]
8. Van Thieu, N.; Barma, S.D.; Van Lam, T.; Kisi, O.; Mahesha, A. Groundwater Level Modeling Using Augmented Artificial Ecosystem Optimization. *J. Hydrol.* **2023**, *617*, 129034. [[CrossRef](#)]
9. Rahnama, M.R.; Abkooh, S.S. Prediction of CO Pollutant in Mashhad Metropolis, Iran: Using Multiple Linear Regression. *Geogr. J.* **2023**, *in press*. [[CrossRef](#)]
10. Sahoo, S.; Jha, M.K. On the Statistical Forecasting of Groundwater Levels in Unconfined Aquifer Systems. *Environ. Earth Sci.* **2015**, *73*, 3119–3136. [[CrossRef](#)]
11. Ebrahimi, H.; Rajaei, T. Simulation of Groundwater Level Variations Using Wavelet Combined with Neural Network, Linear Regression and Support Vector Machine. *Glob. Planet. Chang.* **2017**, *148*, 181–191. [[CrossRef](#)]
12. Bahmani, R.; Ouarda, T.B. Groundwater Level Modeling with Hybrid Artificial Intelligence Techniques. *J. Hydrol.* **2021**, *595*, 125659. [[CrossRef](#)]
13. Poursaeid, M.; Poursaeid, A.H.; Shabanlou, S. A Comparative Study of Artificial Intelligence Models and a Statistical Method for Groundwater Level Prediction. *Water Resour. Manag.* **2022**, *36*, 1499–1519. [[CrossRef](#)]

14. Nia, M.A.; Panahi, F.; Ehteram, M. Convolutional Neural Network- ANN- E (Tanh): A New Deep Learning Model for Predicting Rainfall. *Water Resour. Manag.* **2023**, *37*, 1785–1810. [[CrossRef](#)]
15. Chandar, S.K. Convolutional Neural Network for Stock Trading Using Technical Indicators. *Autom. Softw. Eng.* **2022**, *29*, 16. [[CrossRef](#)]
16. Yang, D.; Karimi, H.R.; Gelman, L. A Fuzzy Fusion Rotating Machinery Fault Diagnosis Framework Based on the Enhancement Deep Convolutional Neural Networks. *Sensors* **2022**, *22*, 671. [[CrossRef](#)] [[PubMed](#)]
17. Yi, C.; Huang, W.; Pan, H.; Dong, J. WLP-VBL: A Robust Lightweight Model for Water Level Prediction. *Electronics* **2023**, *12*, 4048. [[CrossRef](#)]
18. Ghasemlounia, R.; Gharehbaghi, A.; Ahmadi, F.; Saadatnejadgharahassanlou, H. Developing a Novel Framework for Forecasting Groundwater Level Fluctuations Using Bi-Directional Long Short-Term Memory (BiLSTM) Deep Neural Network. *Comput. Electron. Agric.* **2021**, *191*, 106568. [[CrossRef](#)]
19. Khozani, Z.S.; Banadkooki, F.B.; Ehteram, M.; Ahmed, A.N.; El-Shafie, A. Combining Autoregressive Integrated Moving Average with Long Short-Term Memory Neural Network and Optimisation Algorithms for Predicting Ground Water Level. *J. Clean. Prod.* **2022**, *348*, 131224. [[CrossRef](#)]
20. Verma, M.; Ghritlahre, H.K.; Chandrakar, G. Wind Speed Prediction of Central Region of Chhattisgarh (India) Using Artificial Neural Network and Multiple Linear Regression Technique: A Comparative Study. *Ann. Data Sci.* **2023**, *10*, 851–873. [[CrossRef](#)]
21. Al-Qaness, M.A.A.; Ewees, A.A.; Thanh, H.V.; AlRassas, A.M.; Dahou, A.; Elaziz, M.A. Predicting CO₂ Trapping in Deep Saline Aquifers Using Optimized Long Short-Term Memory. *Environ. Sci. Pollut. Res.* **2022**, *30*, 33780–33794. [[CrossRef](#)]
22. Meng, H.; Geng, M.; Han, T. Long Short-Term Memory Network with Bayesian Optimization for Health Prognostics of Lithium-Ion Batteries Based on Partial Incremental Capacity Analysis. *Reliab. Eng. Syst. Saf.* **2023**, *236*, 109288. [[CrossRef](#)]
23. Alizamir, M.; Shiri, J.; Fard, A.F.; Kim, S.; Gorgij, A.D.; Heddami, S.; Singh, V.P. Improving the Accuracy of Daily Solar Radiation Prediction by Climatic Data Using an Efficient Hybrid Deep Learning Model: Long Short-Term Memory (LSTM) Network Coupled with Wavelet Transform. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106199. [[CrossRef](#)]
24. Shi, T.; Liu, Y.; Zheng, X.; Hu, K.; Huang, H.; Liu, H.; Huang, H. Recent Advances in Plant Disease Severity Assessment Using Convolutional Neural Networks. *Sci. Rep.* **2023**, *13*, 2336. [[CrossRef](#)]
25. Moutik, O.; Sekkat, H.; Tigani, S.; Chehri, A.; Saadane, R.; Tchakoucht, T.A.; Paul, A. Convolutional Neural Networks or Vision Transformers: Who Will Win the Race for Action Recognitions in Visual Data? *Sensors* **2023**, *23*, 734. [[CrossRef](#)]
26. Zhang, Q.; Xiao, J.; Tian, C.; Lin, J.C.; Zhang, S. A Robust Deformed Convolutional Neural Network (CNN) for Image Denoising. *CAAI Trans. Intell. Technol.* **2023**, *8*, 331–342. [[CrossRef](#)]
27. Ehteram, M.; Ahmed, A.N.; Khozani, Z.S.; El-Shafie, A. Convolutional Neural Network -Support Vector Machine Model-Gaussian Process Regression: A New Machine Model for Predicting Monthly and Daily Rainfall. *Water Resour. Manag.* **2023**, *37*, 3631–3655. [[CrossRef](#)]
28. Pan, J.-S.; Sun, B.; Chu, S.-C.; Zhu, M.; Shieh, C.-S. A Parallel Compact Gannet Optimization Algorithm for Solving Engineering Optimization Problems. *Mathematics* **2023**, *11*, 439. [[CrossRef](#)]
29. Pang, A.; Liang, H.; Lin, C.; Yao, L. A Surrogate-Assisted Adaptive Bat Algorithm for Large-Scale Economic Dispatch. *Energies* **2023**, *16*, 1011. [[CrossRef](#)]
30. Essa, K.S.; Diab, Z.E. Gravity Data Inversion Applying a Metaheuristic Bat Algorithm for Various Ore and Mineral Models. *J. Geodyn.* **2023**, *155*, 101953. [[CrossRef](#)]
31. Nayak, J.; Swapnarekha, H.; Naik, B.; Dhiman, G.; Vimal, S. 25 Years of Particle Swarm Optimization: Flourishing Voyage of Two Decades. *Arch. Comput. Methods Eng.* **2023**, *30*, 1663–1725. [[CrossRef](#)]
32. Piotrowski, A.P.; Napiorkowski, J.J.; Piotrowska, A.E. Particle Swarm Optimization or Differential Evolution—A comparison. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106008. [[CrossRef](#)]
33. Song, Y.; Wei, L.; Yang, Q.; Wu, J.; Xing, L.; Chen, Y. RL-GA: A Reinforcement Learning-Based Genetic Algorithm for Electromagnetic Detection Satellite Scheduling Problem. *Swarm Evol. Comput.* **2023**, *77*, 101236. [[CrossRef](#)]
34. Let, S.; Bar, N.; Basu, R.K.; Das, S.K. Minimum Elutriation Velocity of the Binary Solid Mixture—Empirical Correlation and Genetic Algorithm (GA) Modeling. *Korean J. Chem. Eng.* **2023**, *40*, 248–254. [[CrossRef](#)]
35. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary Grey Wolf Optimization Approaches for Feature Selection. *Neurocomputing* **2016**, *172*, 371–381. [[CrossRef](#)]
36. Rizk-Allah, R.M.; Hassanien, A.E.; Elhoseny, M.; Gunasekaran, M. A New Binary Salp Swarm Algorithm: Development and Application for Optimization Tasks. *Neural Comput. Appl.* **2019**, *31*, 1641–1663. [[CrossRef](#)]
37. Kalra, V.; Kashyap, I.; Kaur, H. Effect of Ensembling over K-fold Cross-Validation with Weighted K-Nearest Neighbour for Classification in Medical Domain. In Proceedings of the 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, COM-IT-CON 2022, Faridabad, India, 26–27 May 2022. [[CrossRef](#)]
38. Chinchu Krishna, S.; Paul, V. Multi-Class IoT Botnet Attack Classification and Evaluation Using Various Classifiers and Validation Techniques. In *Data Intelligence and Cognitive Informatics*; Jacob, I.J., Kolandapalayam Shanmugam, S., Izonin, I., Eds.; Algorithms for Intelligent Systems; Springer: Singapore, 2023. [[CrossRef](#)]
39. Ehteram, M.; Najah Ahmed, A.; Khozani, Z.S.; El-Shafie, A. Graph Convolutional Network – Long Short Term Memory Neural Network- Multi Layer Perceptron- Gaussian Process Regression Model: A New Deep Learning Model for Predicting Ozone Concentration. *Atmos. Pollut. Res.* **2023**, *14*, 101766. [[CrossRef](#)]

40. Yates, L.A.; Aandahl, Z.; Richards, S.A.; Brook, B.W. Cross Validation for Model Selection: A Review with Examples from Ecology. *Ecol. Monogr.* **2023**, *93*, e1557. [[CrossRef](#)]
41. Şen, Z.; Şişman, E.; Dabanli, I. Innovative Polygon Trend Analysis (IPTA) and applications. *J. Hydrol.* **2019**, *575*, 202–210. [[CrossRef](#)]
42. Awadallah, M.A.; Hammouri, A.I.; Al-Betar, M.A.; Braik, M.S.; Elaziz, M.A. Binary Horse herd optimization algorithm with crossover operators for feature selection. *Comput. Biol. Med.* **2022**, *141*, 105152. [[CrossRef](#)]
43. Li, H.; Lu, Y.; Zheng, C.; Yang, M.; Li, S. Groundwater Level Prediction for the Arid Oasis of Northwest China Based on the Artificial Bee Colony Algorithm and a Back-propagation Neural Network with Double Hidden Layers. *Water* **2019**, *11*, 860. [[CrossRef](#)]
44. Mirmozaffari, M.; Yazdani, M.; Boskabadi, A.; Dolatsara, H.; Kabirifar, K.; Golilarz, N. A Novel Machine Learning Approach Combined with Optimization Models for Eco-Efficiency Evaluation. *Appl. Sci.* **2020**, *10*, 5210. [[CrossRef](#)]
45. Kombo, O.H.; Kumaran, S.; Sheikh, Y.H.; Bovim, A.; Jayavel, K. Long-Term Groundwater Level Prediction Model Based on Hybrid KNN-RF Technique. *Hydrology* **2020**, *7*, 59. [[CrossRef](#)]
46. Huang, J.-Y.; Shih, D.-S. Assessing Groundwater Level with a Unified Seasonal Outlook and Hydrological Modeling Projection. *Appl. Sci.* **2020**, *10*, 8882. [[CrossRef](#)]

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