In the Name of God

Hydroinformatics

Final Project on Neural and Fuzzy Networks

(GMDH, GRNN, and Fuzzy Regression Least Squares Algorithms for Rainfall Estimation)

Group Members:

Mehran Besharatifar

Mohammad Ramezani

Advisor: Dr. Nasseri

Faculty of Civil Engineering

Winter 2023–2024

PROBLEM STATEMENT

Rainfall estimation is crucial for several reasons:

- **Weather Forecasting:** Measuring precipitation is vital for weather forecasting. Forecasts are based on complex mathematical equations in which variables are determined through observations such as the amount and intensity of rainfall.
- **Source of Freshwater:** Rainfall is a source of freshwater. We use water not only for our daily needs but also for energy production.
- **Global Water and Energy Cycle:** Rainfall is an essential component of the global water and energy cycle, which governs climate, weather, and ecological systems.

The purpose of estimating rainfall is to accurately image the precipitation system, especially in areas with poor radar coverage. Incorporating and integrating other data sources such as rain gauges, microwave links, and satellite observations can provide a better picture of the precipitation system. Quantitative rainfall estimation with high resolution and accuracy is a challenging task due to the spatial and temporal variability and the complex microphysics of precipitation processes [1].

Conceptual and physically-based models, although they attempt to account for all the physical processes involved in the rainfall-runoff (R-R) process, are often limited in successful application due to the need for basin-specific parameters and simplifications present in the governing equations [2].

For nearly the past two decades, **Artificial Neural Networks (ANNs)** have emerged as a powerful computational system for highly complex and nonlinear systems. ANNs belong to black-box time series models and offer a relatively flexible and fast modeling tool. Due to their parallel architecture, these models can partially handle the nonlinearity of the system [3]. ANN models provide better predictions compared to traditional models; however, their application is still mostly confined to research environments [4].

In the literature related to ANN applications for simulating R-R processes, many published studies describe the use of ANN models in regions where dataset quality is good and highly reliable. In contrast, the use of ANN models in regions with poor data quality and limited quantity is rarely reported [5].

The present study focuses on the first case, in which sufficient hydrological data is available. Additionally, the data quality is reliable and of relatively good standard. In this study, we aim to estimate the monthly rainfall in the Zayandeh Rood basin by developing **GMDH, GRNN, and Fuzzy Regression Least Squares algorithms**, and evaluate the performance of the developed models using appropriate statistical indicators.

We aim to develop a regression model to estimate rainfall in the Zayandeh Rood watershed using averaged meteorological and hydrological data from the region. The watershed spans an area of 26,864 square kilometers and lies across two provinces: Isfahan and Chaharmahal and Bakhtiari. Approximately 93% of the watershed's area and 98% of its population are situated in Isfahan Province, while the remaining 7% of the area and 2% of the population are located in Chaharmahal and Bakhtiari Province.

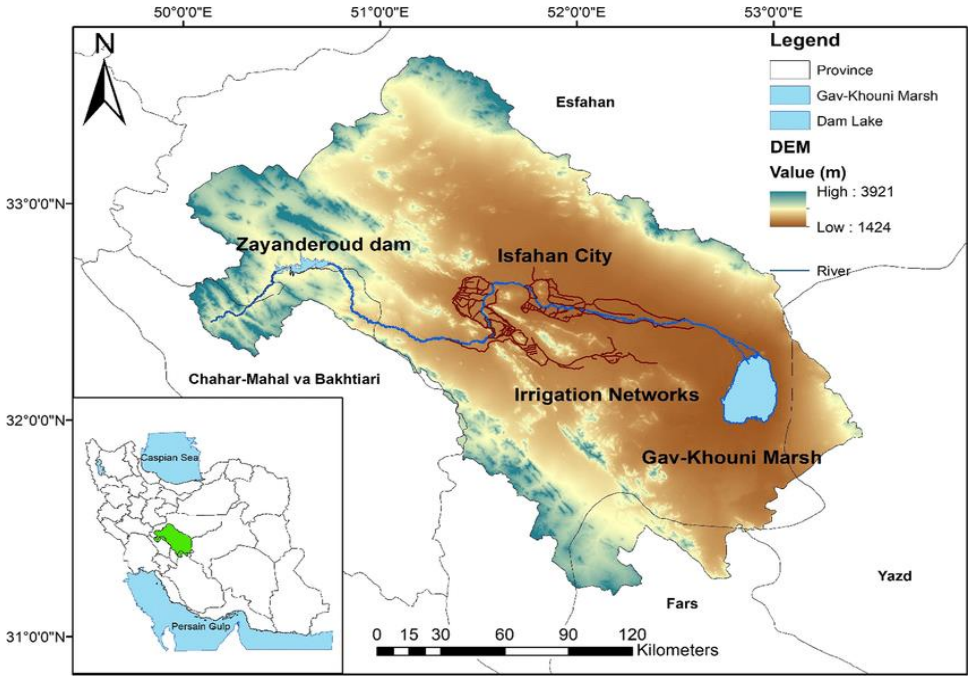


Fig1

All meteorological inputs can be incorporated into the data-driven model to estimate rainfall within the basin. In this report, following the completion of data preprocessing and dimensionality reduction, rainfall estimation is carried out using three algorithms: GMDH, GRNN, and Fuzzy Regression based on Least Squares. After the modeling process, the performance of each model is evaluated using statistical metrics such as R^2 and RMSE. Finally, a comprehensive summary of the models' performance is provided.

Evaluation Statistics

The statistical metrics used in this study are as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}{N}} \quad (1)$$

- Root Mean Square Error (RMSE): Quantifies the difference between observed and simulated values.

- Nash-Sutcliffe Efficiency (NSE): Proposed by Nash and Sutcliffe (1970), this metric measures how well the model reflects observed changes.

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{n} (\bar{y}_i - y_i)^2}{\sum_{i=1}^{n} (y_i - y)^2} \tag{2}$$

- Kling-Gupta Efficiency (KGE): Proposed by Gupta et al. (2009), this metric provides a comprehensive assessment of the model's similarity to observed data.

$$\mathrm{KGE} = 1 - \sqrt{(R(\bar{y}_i \cdot y_i) - 1)^2 + \left(\frac{\mathrm{var}\ (\bar{y}_i)^{1/2}}{\mathrm{var}\ (y_i)^{1/2}} - 1\right)^2 + \left(\frac{\bar{y}'}{y} - 1\right)^2} \tag{3}$$

- Bias: Identifies any systematic error in the model.

$$\mathrm{Bias} = \frac{1}{n}\sum_{i=1}^{n} (\bar{y}_i - y_i) \tag{4}$$

In equations (1) to (4) utilized to evaluate the accuracy of the methods, y_i denotes the observed value, $\hat{y}_i$ represents the simulated value, $\bar{y}$ is the mean of the observed values, and $\hat{y}'$ indicates the simulated mean at time i. Additionally, R and var denote the correlation coefficient (CC) and variance, respectively.

GENERAL REGRESSION NEURAL NETWORK (GRNN)

GENERAL REGRESSION NEURAL NETWORKS (GRNN)

At the initial stage, a frequency distribution chart of the target data is plotted to gain a deeper understanding of the dataset. As shown, the majority of the target values are concentrated around numbers close to zero.

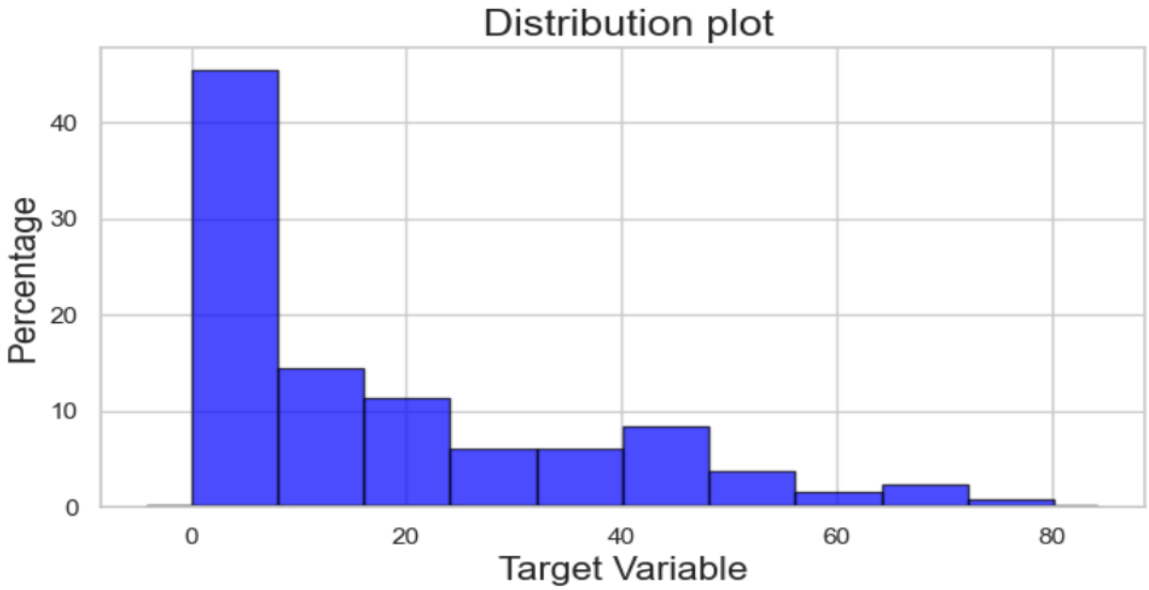


Fig2

Feature extraction using Principal Component Analysis (PCA)

At this stage, extracting the principal components of the dataset is essential for reducing dimensionality and improving model efficiency. We apply the Principal Component Analysis (PCA) technique to identify the directions in which the data varies the most. This involves transforming the original set of correlated variables into a smaller set of uncorrelated components, each capturing a portion of the dataset's total variance.

To better understand the contribution of each component, we calculate and plot the **explained variance ratio** for every principal component. This visualization helps determine how many components should be retained without significantly losing information. According to the resulting chart, the **first principal component alone accounts for approximately 73%** of the total variance, indicating that it carries a substantial amount of the dataset's information. Therefore, using a reduced number of components can still preserve the essential structure of the data while reducing computational complexity.

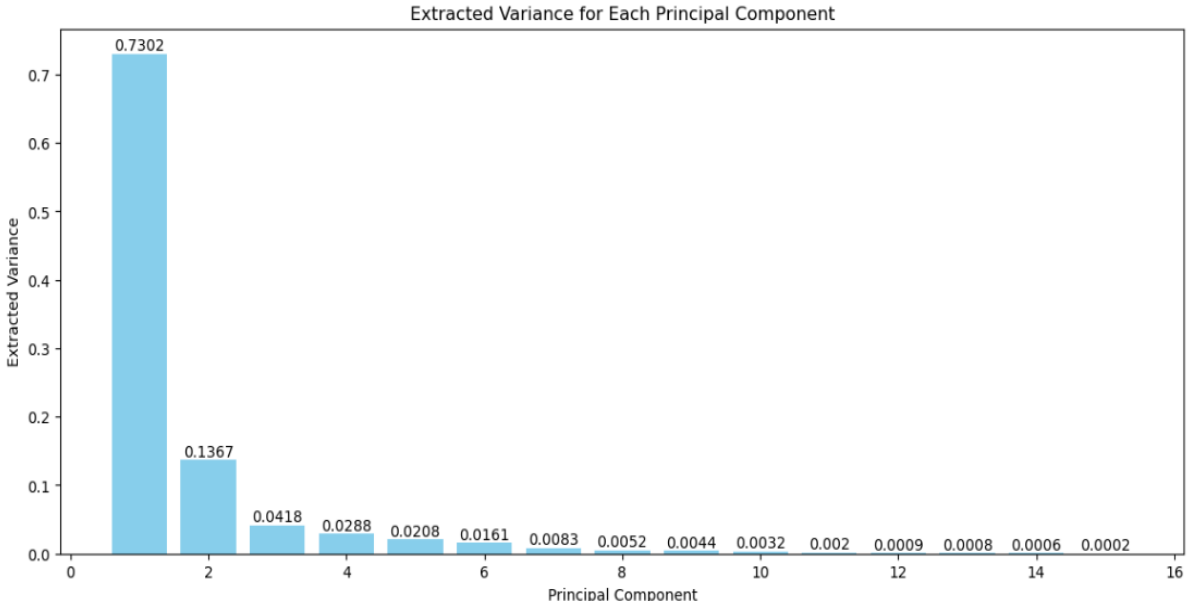


Fig3

In Principal Component Analysis (PCA), dimensionality reduction involves transforming the original variables into a new set of uncorrelated variables called principal components. Each principal component is a **linear combination** of the original variables, and the **loading values** represent the weights or coefficients of these original variables in the linear combination. These loading values indicate the contribution of each variable to a given principal component, thereby revealing how much influence each feature has on the component's formation.

The chart below displays the **loading values for each principal component**, providing valuable insight into which variables are most influential. This visualization plays a critical role in **interpreting the components** and selecting the most meaningful ones for further modeling.

.

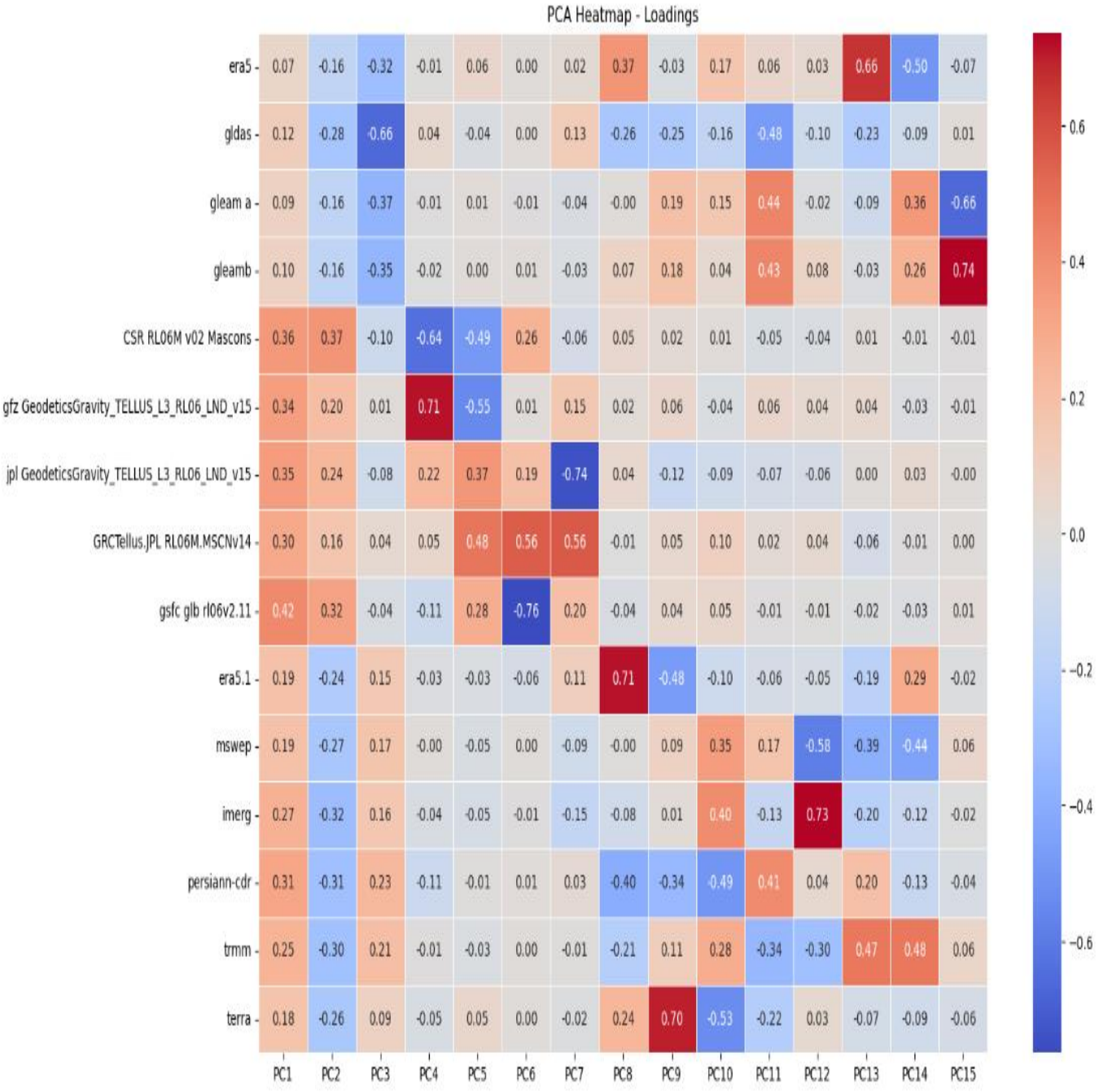

PCA Heatmap - Loadings
era5 0.07 -0.16 -0.32 -0.01 0.06 0.00 0.02 0.37 -0.03 0.17 0.06 0.03 0.66 -0.50 -0.07
gldas 0.12 -0.28 -0.66 0.04 -0.04 0.00 0.13 -0.26 -0.25 -0.16 -0.48 -0.10 -0.23 -0.09 0.01
gleam a 0.09 -0.16 -0.37 -0.01 0.01 -0.01 -0.04 -0.00 0.19 0.15 0.44 -0.02 -0.09 0.36 -0.66
gleamb 0.10 -0.16 -0.35 -0.02 0.00 0.01 -0.03 0.07 0.18 0.04 0.43 0.08 -0.03 0.26 0.74
CSR RL06M v02 Mascons 0.36 0.37 -0.10 -0.64 -0.49 0.26 -0.06 0.05 0.02 0.01 -0.05 -0.04 0.01 -0.01 -0.01
gfz GeodeticsGravity_TELLUS_L3_RL06_LND_v15 0.34 0.20 0.01 0.71 -0.55 0.01 0.15 0.02 0.06 -0.04 0.06 0.04 0.04 -0.03 -0.01
jpl GeodeticsGravity_TELLUS_L3_RL06_LND_v15 0.35 0.24 -0.08 0.22 0.37 0.19 -0.74 0.04 -0.12 -0.09 -0.07 -0.06 0.00 0.03 -0.00
GRCTellus.JPL RL06M.MSCNv14 0.30 0.16 0.04 0.05 0.48 0.56 0.56 -0.01 0.05 0.10 0.02 0.04 -0.06 -0.01 0.00
gsfc glb rl06v2.11 0.42 0.32 -0.04 -0.11 0.28 -0.76 0.20 -0.04 0.04 0.05 -0.01 -0.01 -0.02 -0.03 0.01
era5.1 0.19 -0.24 0.15 -0.03 -0.03 -0.06 0.11 0.71 -0.48 -0.10 -0.06 -0.05 -0.19 0.29 -0.02
mswep 0.19 -0.27 0.17 -0.00 -0.05 0.00 -0.09 -0.00 0.09 0.35 0.17 -0.58 -0.39 -0.44 0.06
imerg 0.27 -0.32 0.16 -0.04 -0.05 -0.01 -0.15 -0.08 0.01 0.40 -0.13 0.73 -0.20 -0.12 -0.02
persiann-cdr 0.31 -0.31 0.23 -0.11 -0.01 0.01 0.03 -0.40 -0.34 -0.49 0.41 0.04 0.20 -0.13 -0.04
trmm 0.25 -0.30 0.21 -0.01 -0.03 0.00 -0.01 -0.21 0.11 0.28 -0.34 -0.30 0.47 0.48 0.06
terra 0.18 -0.26 0.09 -0.05 0.05 0.00 -0.02 0.24 0.70 -0.53 -0.22 0.03 -0.07 -0.09 -0.06
PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10 PC11 PC12 PC13 PC14 PC15
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Fig4

GRNN:

The determination of the **spread** is the only degree of freedom in this type of powerful neural network, and it must be defined. This is handled by the algorithm itself. Next, we define the **GRNN network**. Approximately **70 to 80 percent** of the data is allocated for **training**, **10 to 15 percent** for **validation**, and the remaining percentage is used for **testing**.

Next step: Running the model with optimized parameters and evaluating it — After finding the optimal spread value, we can run the model on the main dataset and calculate the model's **performance evaluation metrics**.

.

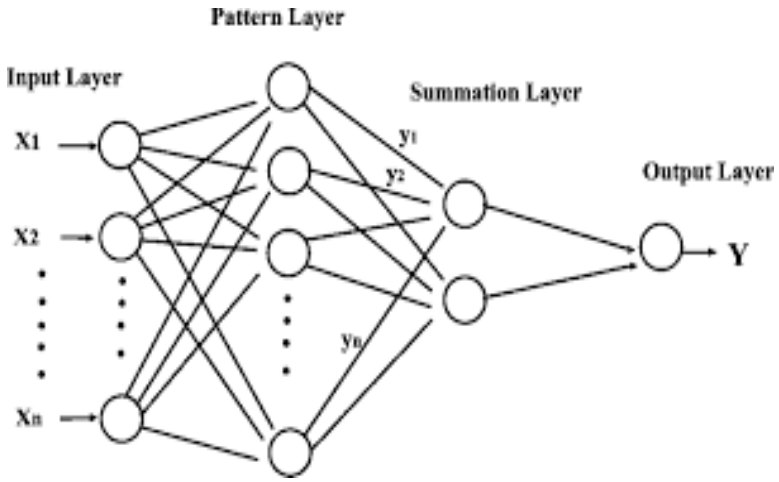


Fig5

RESULTS AND EVALUATION

As we can see, the output results from the model are presented as follows:

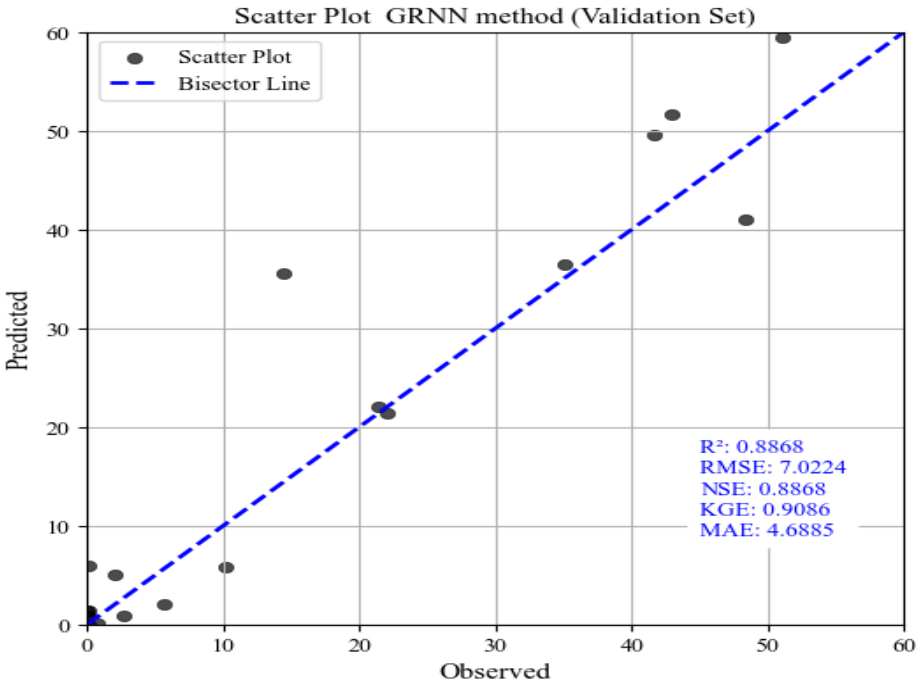


Fig6. For model evaluation, **5 similarity and dissimilarity indices** have been used.

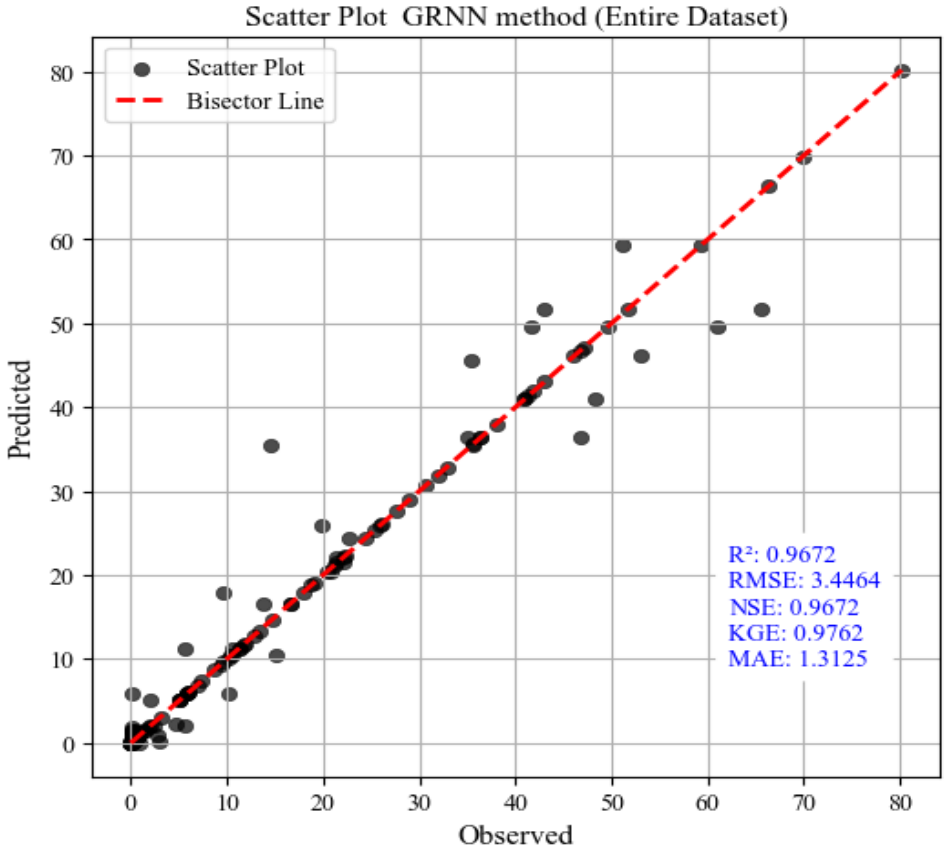


Fig7. For model evaluation, **5 similarity and dissimilarity indices** have been used.

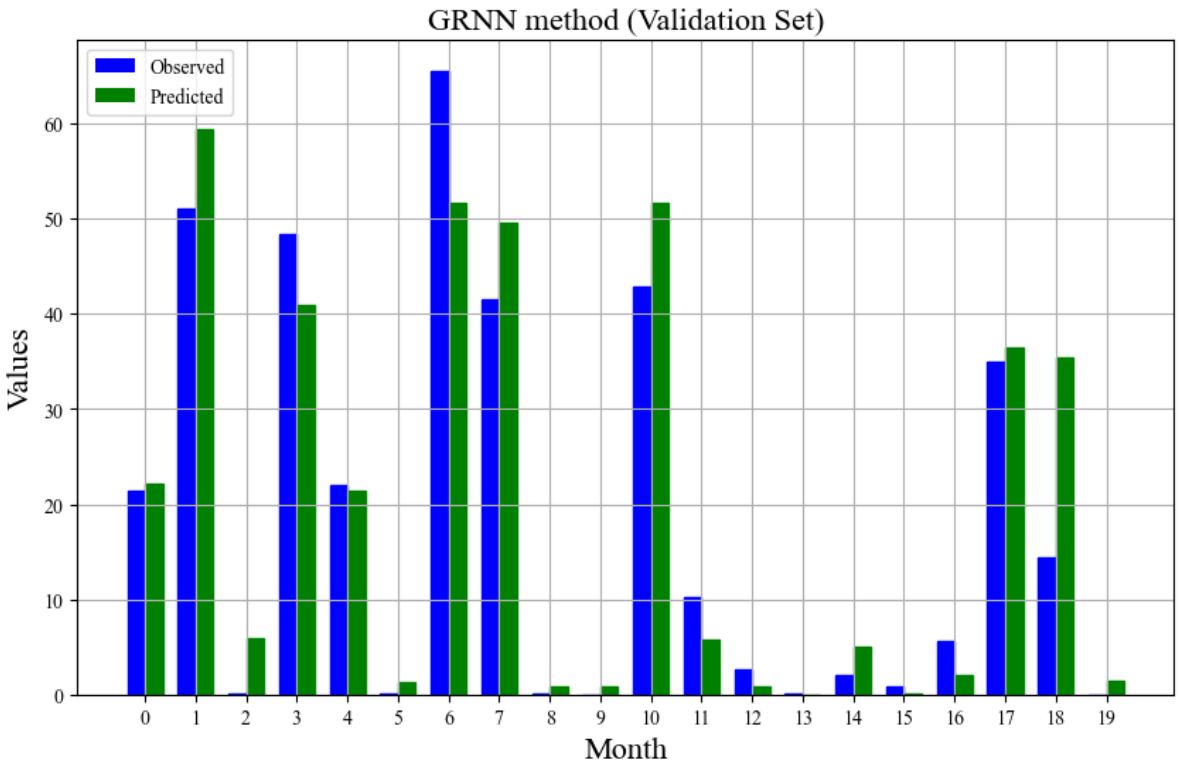
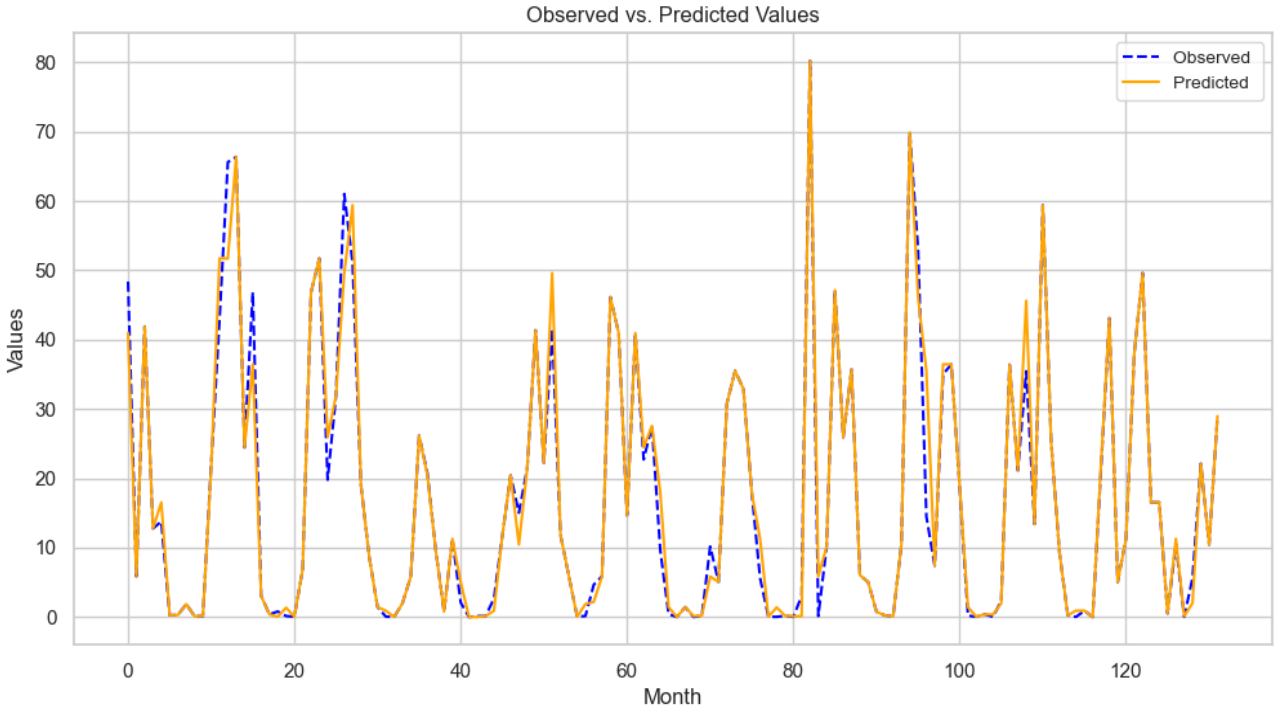


Fig8. For model evaluation, **5 similarity and dissimilarity indices** have been used.

Fig9. For model evaluation, **5 similarity and dissimilarity indices** have been used.



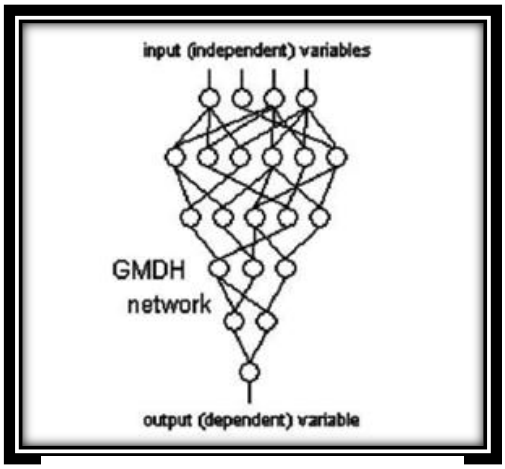
GROUP METHOD OF DATA HANDLING
(GMDH)

GMDH network

The **Group Method of Data Handling (GMDH)** is a self-organizing, one-way neural network that allows for the modeling of a complex nonlinear system using a multilayer neural network. This network consists of multiple layers, and each layer is made up of several neurons. The goal of this type of network is to determine the function $\hat{f}$ using **M sets of multi-input and single-output data**, in such a way that for an input vector $X = [x_1, x_2, x_3, \dots x_n]$, the predicted output $\hat{y}$ is as close as possible to the actual value y, thereby minimizing the error function between the actual output and the predicted output:

$$\sum_{k=1}^{M} \left[\hat{f}\ (x_{k1}, x_{k2}, x_{k3}, \dots x_{kn}) - y_k\right]^2$$

The figure below shows the structure of a general GMDH neural network using **two-variable polynomial functions**.



According to this structure, the first layer (at the top) provides an input for each predictor variable. Each neuron in the second layer is activated by inputs from two input variables. Neurons in the third layer are activated by inputs from two neurons in the previous layer, and this process repeats for all layers. The final layer (at the bottom) takes its two inputs from the previous layer and produces a value, which is the network's output. The mathematical foundation of this network is based on the decomposition of the Volterra series into **second-degree two-variable polynomials**, as shown in equation 7:

$$G(x_m, x_n) = a_0 + a_1 x_m + a_2 x_n + a_3 x_m x_n + a_4 + a_5 x_n^2$$

A set of polynomials is constructed using the above relation, and the unknown coefficients of all of them are obtained using the **Least Squares (LS) method**. For each function (each constructed neuron), the coefficients of the equations for each neuron are derived to minimize the total error in order to optimally fit the inputs across all input-output pair datasets.

RESULTS AND EVALUATION

As we can see, the output results from the model are presented as follows:

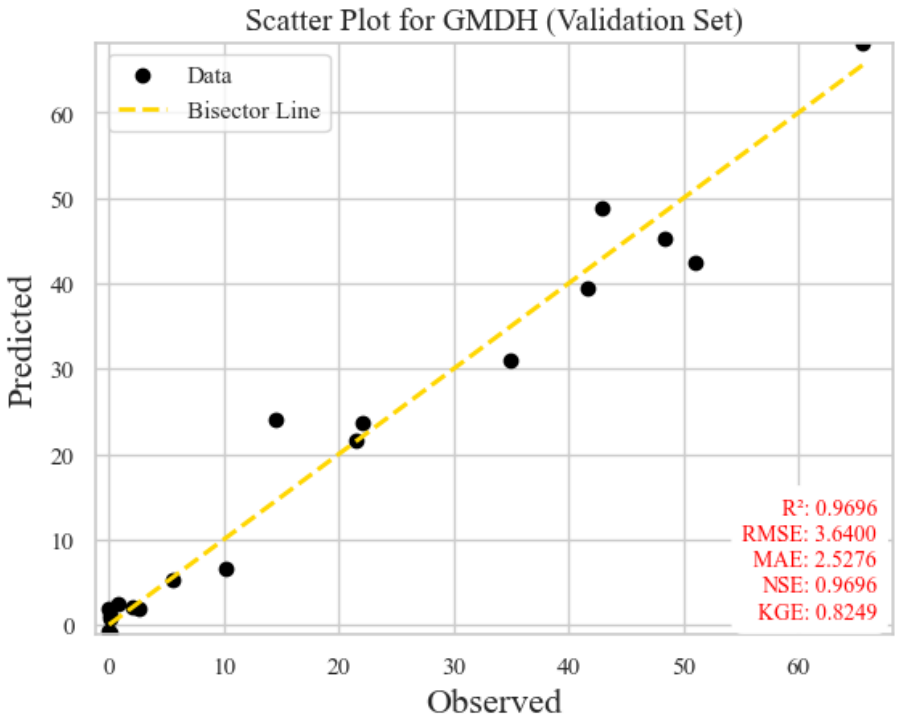


Fig10. For model evaluation, **5 similarity and dissimilarity indices** have been used.

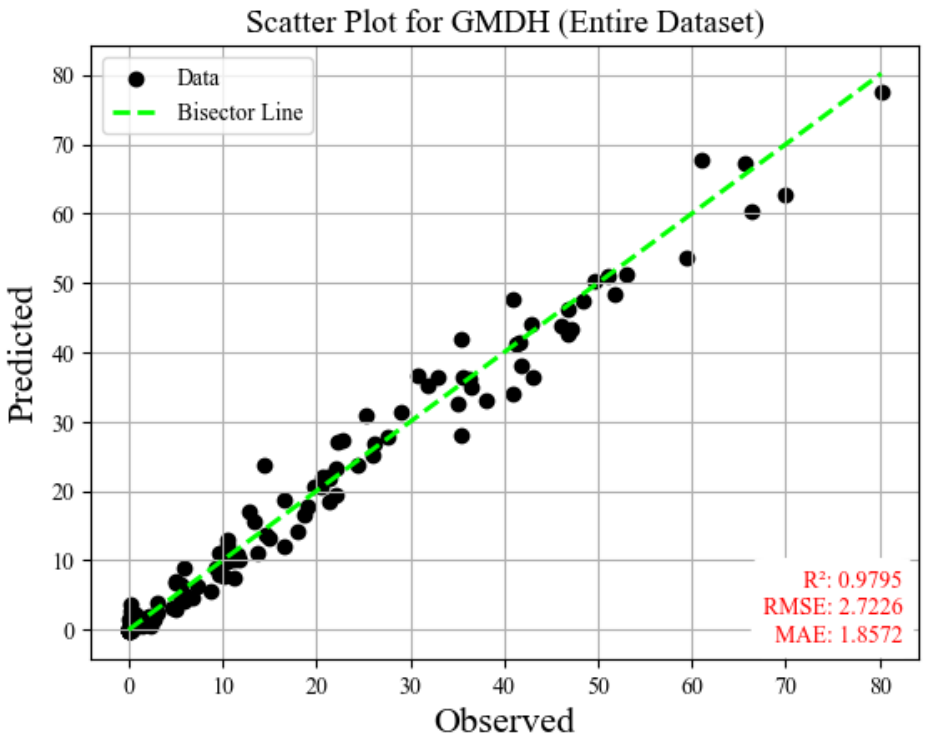

Scatter Plot for GMDH (Entire Dataset)
Data
Bisector Line
Predicted
Observed
R²: 0.9795
RMSE: 2.7226
MAE: 1.8572

Fig11

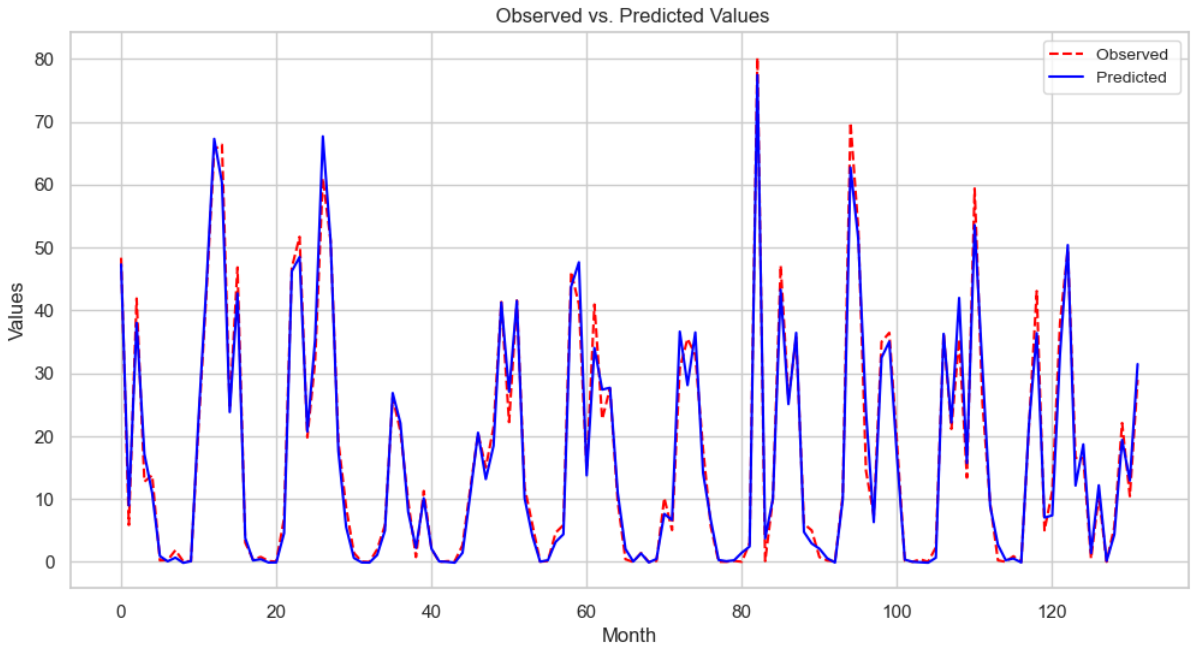

Observed vs. Predicted Values
Observed
Predicted
Values
Month

Fig12

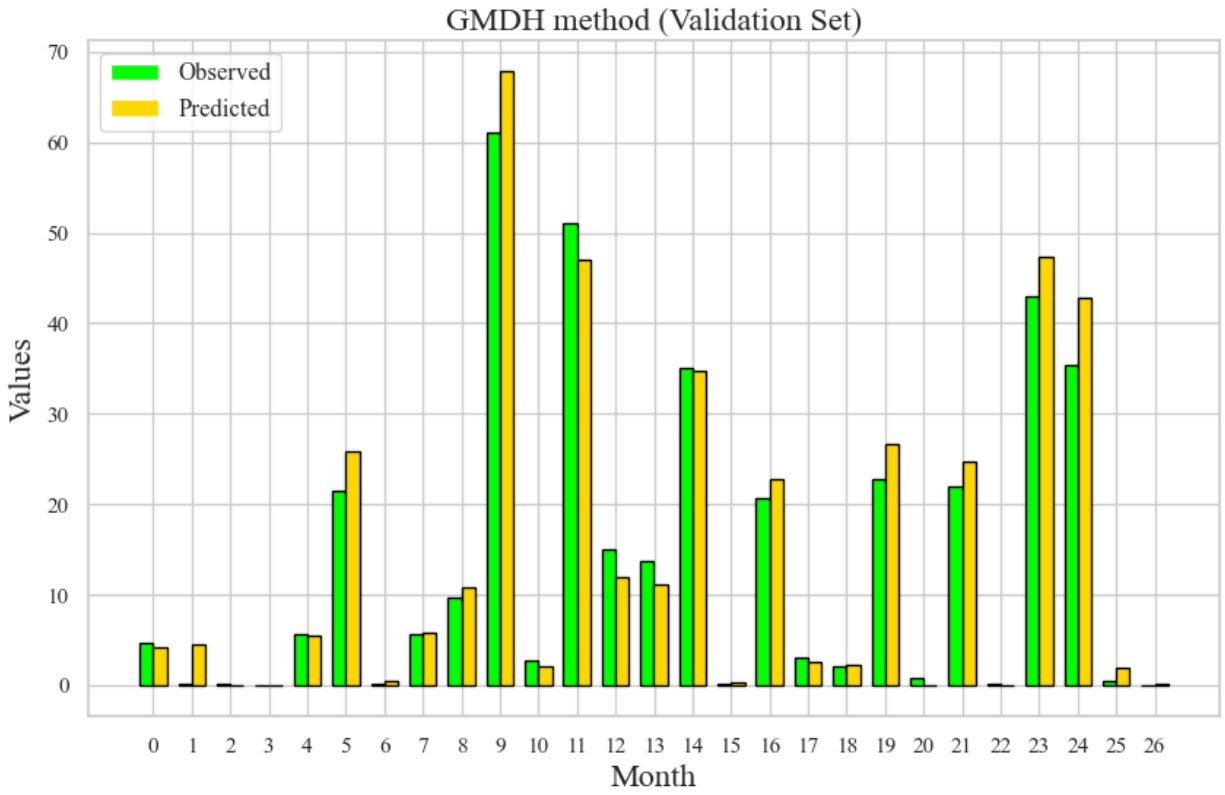


Fig13.The predicted and observed values have been examined in the charts above.

FUZZY LEAST-SQUARES LINEAR REGRESSION

Fuzzy Regression Using Least Squares

This algorithm is a statistical technique that combines fuzzy logic with regression analysis. Fuzzy logic deals with uncertainty and imprecision in data, while regression analysis is a statistical method used to model the relationship between a dependent variable and one or more independent variables. Fuzzy regression enables modeling of relationships in cases where the data involved are imprecise or uncertain. Below is a brief overview of how fuzzy regression with the least squares approach is conducted:

Fuzzy Sets and Membership Functions:

In fuzzy regression, variables are represented as fuzzy sets, and membership functions describe the degree to which each data point belongs to a fuzzy set. This is particularly useful when dealing with ambiguous or inaccurate information.

Fuzzy Regression Model:

The fuzzy regression model is formulated based on fuzzy sets and their membership functions. It defines the relationship between the dependent variable and one or more independent variables in a fuzzy environment. The objective is to identify the model parameters that minimize error using the least squares method.

Least Squares Estimation:

The least squares approach is used to estimate the parameters of the fuzzy regression model. The goal is to minimize the sum of squared differences between observed values and predicted values. This minimization process is adapted to accommodate fuzzy relationships and uncertainties in the data.

Optimization:

Optimization algorithms, often tailored for fuzzy systems, are used to find the optimal parameter values in the fuzzy regression model. This involves adjusting the parameters to minimize overall error.

Inference and Prediction:

Once the fuzzy regression model is built, it can be used for inference and prediction. The model provides insights into the relationships among variables and can be used to make predictions for new data points.

Fuzzy regression using least squares is particularly applicable in situations where traditional regression models may be unsuitable due to uncertainty, imprecision, or vagueness in the data. It is often used in fields where qualitative or subjective information needs to be incorporated into quantitative models.

Implementation Note: Since a specific library for this algorithm is not available in Python by default, the required functions need to be implemented manually.

```python
# Fuzzy logic membership functions
def create_membership_functions(data, num_clusters=3):
    membership_functions = []
    for i in range(data.shape[1]):
        centroids = np.linspace(data[:, i].min(), data[:, i].max(), num_clusters)
        membership_functions.append(centroids)
    return membership_functions

# Create fuzzy membership functions
membership_functions = create_membership_functions(X_train_scaled)

# Fuzzify the input data
def fuzzify_data(data, membership_functions):
    fuzzy_data = []
    for i in range(data.shape[1]):
        fuzzy_data.append(np.array([np.exp(-(point[i] - centroid)**2) for point in data for centroid in membership_functions[i]]).reshape(-1, len(members
    return np.hstack(fuzzy_data)

X_train_fuzzy = fuzzify_data(X_train_scaled, membership_functions)
X_test_fuzzy = fuzzify_data(X_test_scaled, membership_functions)

# Least squares optimization
def least_squares_optimization(X, y):
    weights = np.linalg.lstsq(X, y, rcond=None)[0]
    return weights

# Train the FPLSM model
weights = least_squares_optimization(X_train_fuzzy, y_train)

# Make predictions on the test set
y_pred = np.dot(X_test_fuzzy, weights)
```

The remaining steps follow a similar procedure to the previously discussed algorithms, including model training, parameter optimization, model evaluation, and generating the output results.

Results and Evaluation

As observed, the model results are presented as follows:

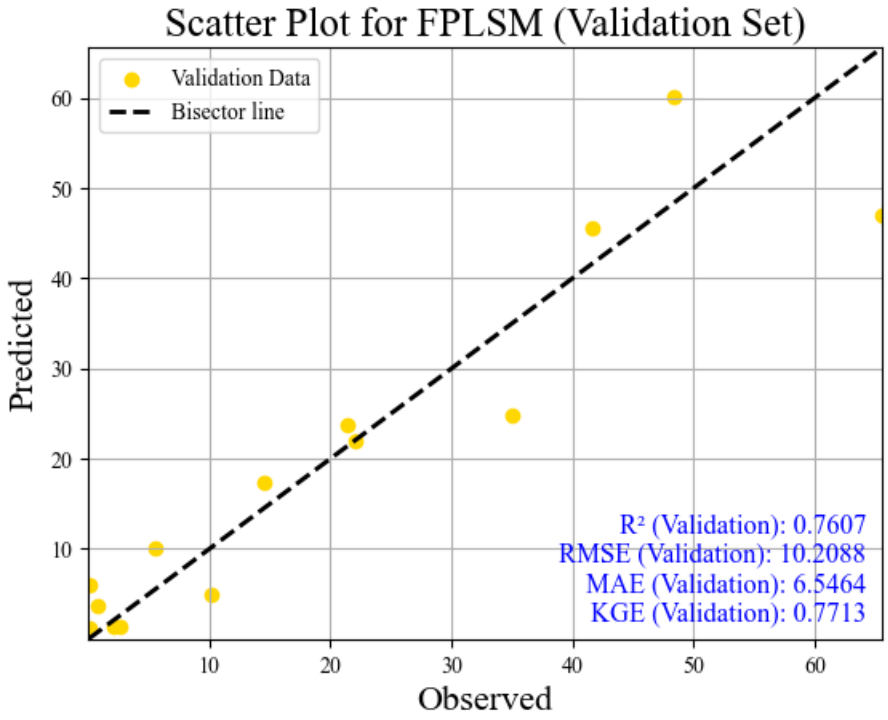


Fig14. Four similarity and dissimilarity indices have been used for model evaluation.

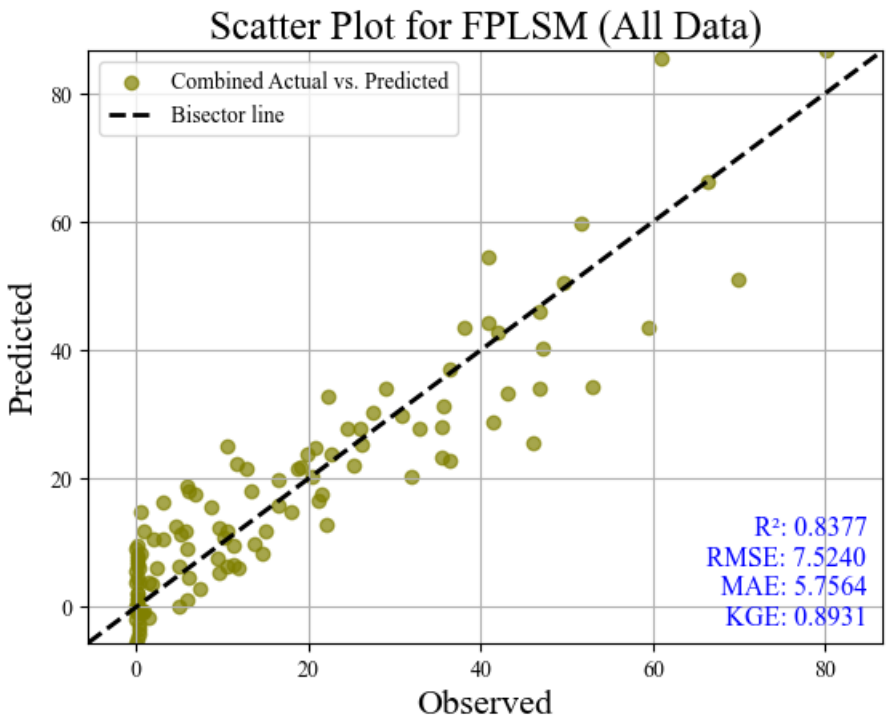


Fig15

You can also observe the fuzzy regression charts using the least squares method. The prediction intervals represent the fuzzy outer boundaries. Since we used 15 features, an equivalent number of charts for the outputs was generated. In fact, this type of chart representation illustrates the degree of correlation between each feature and the target variable.

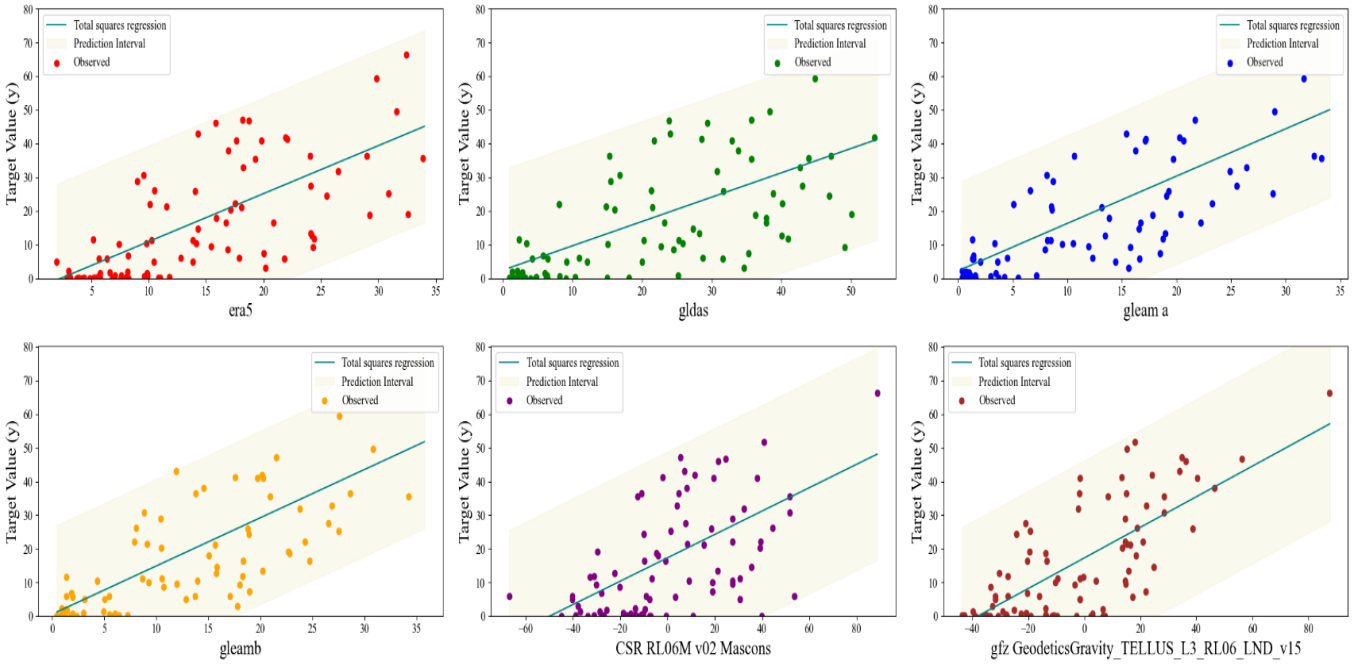

Total squares regression
Prediction Interval
Observed
Target Value (y)
era5
gldas
gleam a
gleamb
CSR RL06M v02 Mascons
gfz GeodeticsGravity_TELLUS_L3_RL06_LND_v15

Fig16

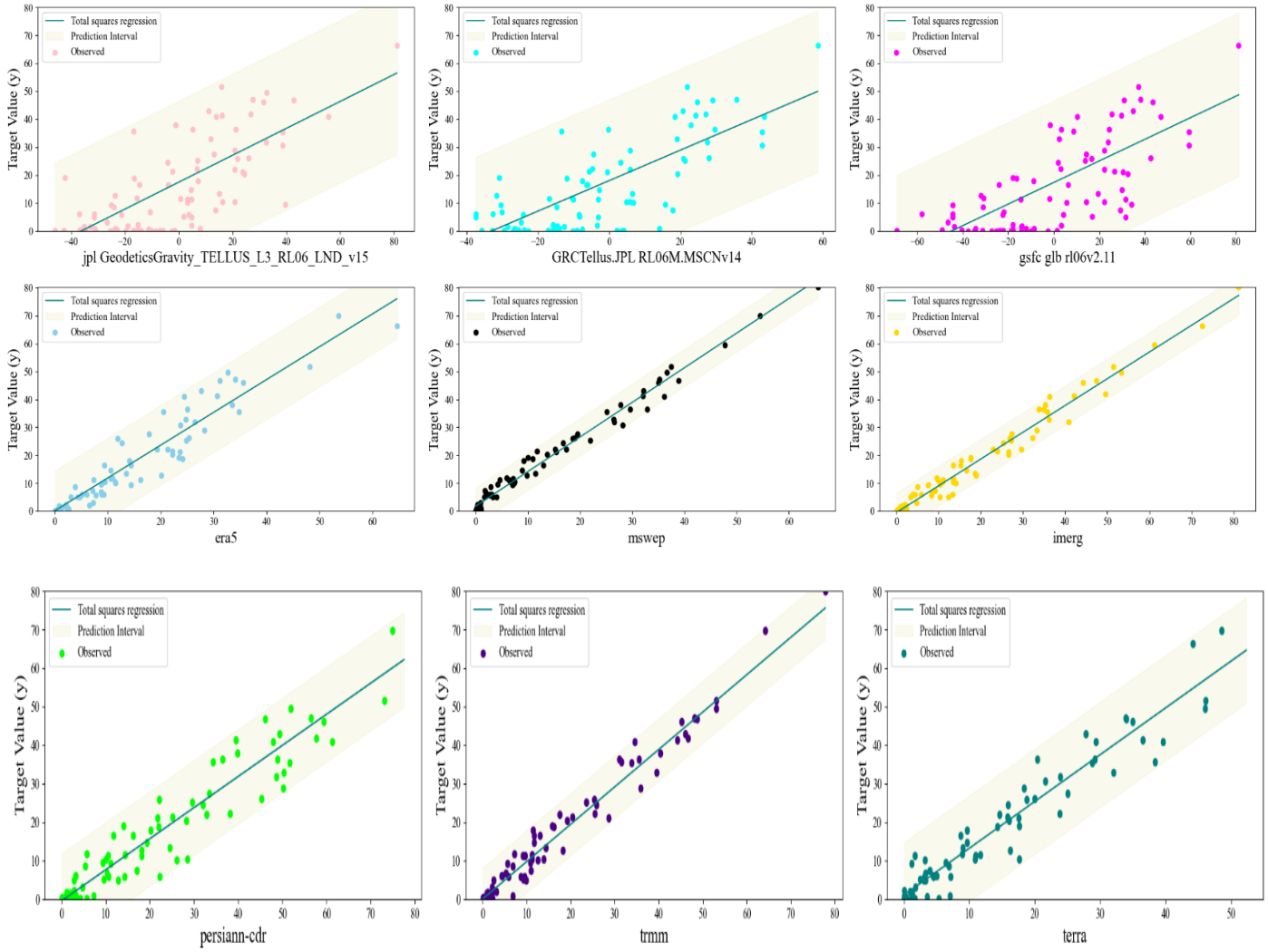

Total squares regression
Prediction Interval
Observed
Target Value (y)
jpl GeodeticsGravity_TELLUS_L3_RL06_LND_v15
GRCTellus.JPL RL06M.MSCNv14
gsfc glb rl06v2.11
era5
mswep
imerg
persiann-cdr
trmm
terra

Fig17

CONCLUSION

In this study, three different methods—FPLSM, GRNN, and GMDH—were utilized for rainfall modeling in the Zayandeh Rood watershed. The data preprocessing phase involved applying Principal Component Analysis (PCA) to reduce the dimensionality of the dataset, enabling a more efficient modeling process by focusing on the most important components of the data. This step helped in addressing issues related to multicollinearity and overfitting, ensuring that the models could generalize better to unseen data.

Following the preprocessing, the three models (FPLSM, GRNN, and GMDH) were developed and trained using the processed data. Each model was evaluated based on its ability to predict rainfall, with performance assessed using a variety of statistical metrics, including R^2 and RMSE.

The results of the validation process indicated that the GMDH algorithm outperformed the other methods in terms of predictive accuracy. The performance of the models was analyzed, and the GMDH approach showed the best capability in capturing the underlying patterns of the rainfall data.

This suggests that GMDH could be a particularly effective method for modeling rainfall in the Zayandeh Rood watershed. Future work could focus on refining the model parameters further or exploring hybrid approaches that combine the strengths of these algorithms to improve predictive performance.

Model	**R2**	**RMSE**	**MAE**	**KGE**
GRNN	**0.88**	**7.02**	**4.68**	**0.90**
GMDH	**0.96**	**3.64**	**2.52**	**0.82**
FPLSM	**0.76**	**10.20**	**6.54**	**0.77**

References

[1] A variable spread fuzzy linear regression model with higher explanatory power and forecasting accuracy. https://doi.org/10.1016/j.ins.2008.06.005

[2] Application of the group method of data handling (GMDH) approach for landslide susceptibility zonation using readily available spatial covariates. https://doi.org/10.1016/j.catena.2021.105779

[3] E. Gaume, R. Gosset, Over-parameterization, a major obstacle to the use of artificial neural networks in hydrology, Hydrology and Earth System Sciences 7 (5) (2003) 693–706.

[4] C.E. Imrie, S. Durucan, A. Korre, River flow prediction using artificial neural networks: generalization beyond the calibration range, Journal of Hydrology (2000) 138–153.

[5] A. Celminš, Least squares model fitting to fuzzy vector data.https://doi.org/10.1016/0165-0114(87)90070-4

[6] Rainfall Runoff Modelling Using Generalized Neural Network and Radial Basis Network.

[7] A combined rotated general regression neural network method for river flow forecasting. https://doi.org/10.1080/02626667.2014.944525

[8] Using the General Regression Neural Network Method to Calibrate the Parameters of a Sub-Catchment. https://doi.org/10.3390/w13081089

[9] Prediction of partition coefficients of alkaloids in ionic liquids based aqueous biphasic systems using hybrid group method of data handling (GMDH) neural network
https://doi.org/10.1016/j.molliq.2013.11.033