INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Combining the Group Method of Data Handling with Firefly Algorithm for Rainfall Forecasting

Azlina Narawi
Faculty of Computing, Universiti Teknologi Malaysia,
81310 UTM Johor Bahru, Johor, Malaysia
Universiti Teknologi MARA Cawangan Sarawak, Kampus Samarahan 2, 94300 Kota Samarahan Sarawak, Malaysia
Email: azlina_14@uitm.edu.my

Dayang N.A. Jawawi*
Faculty of Computing, Universiti Teknologi Malaysia,
81310 UTM Johor Bahru
Johor, Malaysia
Email: dayang@utm.my

Submitted: 30/9/2025. Revised edition: 14/4/2026. Accepted: 4/5/2026. Published online: 10/6/2026
DOI: https://doi.org/10.11113/ijic.v16n1.677

***Abstract*—Rainfall forecasting continues with the evolution of mixed and new research, where a variety of the same models have been suggested to make some improvements in prediction validity. However, at the same time there is a lack of models that seems to be single capable of handling all features of the data to the best of its ability, with multi-method selection difficult. One way to overcome this constraint is that model combination methods have rapidly emerged in time series forecasting, especially precipitation forecasting. These can be an array of simple but complex model pairs. This study proposes a rainfall forecasting framework for the work based on Group Method of Data Handling (GMDH) and Firefly Algorithm (FA). Forecasting is first of all done based on the GMDH (with polynomial transfer function, GMDH-Poly). Then the model is further enlarged with transfer functions (Sigmoid function, Tangent function, and Radial Basis Function). After that, FA is used to heuristically update the weights of each GMDH variant, taking into account the output from this optimisation for a better prediction. This proposed approach is applied in the case of rainfall data in Kuching, Sarawak for the years 2010–2019. The experimental results clearly show the hybrid GMDH-FA model significantly outperforms the standard GMDH model. In particular, GMDH-FA scoring RMSE, MAE and MAPE is 0.0550, 0.0455 and 24.0733 respectively, which is significantly lower than that of, GMDH-Poly (0.0981, 0.0761 and 47.3864) in testing data set. As noted, the massive decrease in error values implies the GMDH-FA model accurately predicts and generalizes well with the newly observed values. Thus, such findings verify the effectiveness of the optimization based weighting approach and the built-in GMDH-FA model is a suitable method as a further rainfall forecasting technique.**

***Keywords*—Group method of data handling, GMDH-FA, forecasting**

I. INTRODUCTION

Since timely and precise forecasts of severe weather can help minimize casualties and damage from natural disasters, effective rainfall forecasting has been one of the most crucial concerns in hydrological study. To increase resistance to the rising frequency and intensity of rainfall extremes, planning adaptation and mitigation strategies are required (Gao *et al*., 2018). Since rainfall data is typically not totally linear or nonlinear, a method to accommodate this feature is being actively investigated. The underlying nonlinear relationship of the data is intricate and challenging to understand. As a result, contemporary research on rainfall forecasting is still ongoing.

Many researchers have utilized a wide range of rainfall forecasting techniques. This empirical strategy is based on an examination of historical rainfall data and its correlation with other meteorological variables. However, a self-organizing model known as the group model of data handling (GMDH) has received a lot of interest recently for the prediction of rainfall. Rainfall forecasting was done using a variety of methodologies.

The GMDH model was introduced by Ivakhnenko, a Russian mathematician in 1968 (Ivakhnenko, 1971). It is a self-organizing heuristic AI model. It can structure its networks without the need for human involvement. Prof. Alexey G. Ivakhnenko first proposed the GMDH model in the 1960s to detect nonlinear relationships between input and output variables, as well as to analyse complex systems, recognise patterns, cluster data, and predict outcomes. According to Tamura and Kondo (1978), the GMDH model

is a powerful practical tool for identifying non-linear input-output interactions in complex systems. Setting parameters, such as the number of neurons and layers, is viewed as one of the more difficult aspects of executing the Artificial Neural Network (ANN) model. The GMDH model, on the other hand, can self-organize its structure throughout the training method; the number of neurons in a layer and the number of layers is unrelated (Tauser & Buryan, 2011). The highest layers number, highest neurons number, and selection pressure must all be established before predicting in the GMDH model (Gao *et al*., 2018). The ANN is prone to overfitting, but the GMDH is resistant to overfitting (Tauser & Buryan, 2011). Due to its ability to automatically self-organize without human intervention, the GMDH can avoid biases and misjudgements allowing it to find the best solution for a given situation (Tauser & Buryan, 2011). The GMDH model can produce explicit expressions, whereas the ANN model can only produce implicit structures. This distinctive feature of the GMDH model aids in the identification of the most influential factors and has been used as an input selection in the past (Ugrasen *et al*., 2014).

The benefits of the GMDH model motivate us to use it for rainfall forecasting. Rainfall forecasting is still a hot topic of research, with researchers employing techniques such as artificial neural networks, genetic algorithms, support vector regression, particle swarm optimization, and fuzzy logic (Shahid *et al*.,20, Danandeh, 2020, Janarthanan, 2021, Wu & Zhou, 2009, and Refonaa *et al*., 2019). Onwubolu *et al*. (2007), on the other hand, used the GMDH model to forecast rainfall. For daily pressure, daily temperature, and monthly rainfall, they disseminate the self-organizing enhanced GMDH (e-GMDH) model. The e-GMDH improved the polynomial GMDH by adding thresholding methods, coefficient rounding, and pruning using a half-randomized selection approach. For daily temperature, the e-GMDH outperformed the polynomial neural network and the enhanced polynomial neural network, according to the findings. In comparison to the other approaches, the e-GMDH did not perform well for monthly rainfall and daily temperature.

The polynomial transfer function is used in Ivakhnenko's traditional GMDH model. Despite this, Jirina (1994) proposed a logistic sigmoid transfer function to re-place the polynomial function in order to solve the multicollinearity generated by the GMDH model's quadratic polynomial. Furthermore, Kondo and Pandya (2003) revealed that the GMDH model that utilizes the sigmoid transfer function performed better than the GMDH model that uses the radial basis function and ANN based on their experimental results. They stated that utilizing sigmoid as the transfer function aids in the identification of nonlinear systems, as the neural network architecture is used to fit complex nonlinear systems (Kondo & Pandya, 2003). Tauser and Buryan (2011), on the other hand, integrated eight alternative transfer functions in their GMDH network, including polynomial, harmonic (cosine), square root, inverse polynomial, logarithmic, arc tangent, and exponential. In fore-casting rainfall in Sarawak, (Narawi *et al*., 2022) examined the polynomial and sigmoid transfer functions for GMDH. The polynomial function dominated the results performance, according to the study. In GMDH-type neural networks, sigmoid function, RBF, and polynomial function were mostly employed to describe the relationship between inputs and outputs (Nourani & Komasi, 2013).

On other hand, nature-inspired metaheuristic algorithms such as Firefly Algorithm (FA), Particle Swarm Optimization (PSO), Bat Algorithm (BA) and others have been popularly used by many researchers. This is due to their ability to resolve optimization problems. FA was introduced by Yang (2008). Yang (2010) and Yang *et al*. (2013) presented a population-based search method, the FA, which was motivated by the firefly's flashing behavior. Non-convex and nonlinear optimization issues have been successfully addressed using FA (Yang, 2010, Yang *et al*., 2013). Recent research has shown that FA is approachable and performs better than other metaheuristic methods. FA is superior to the other three algorithms, according to comparisons between it and PSO, bacteria foraging (BF), and artificial bee colony (ABC) algorithms (Wu & Zhou, 2009). FA will therefore also be combined with GMDH in this work in order to enhance the performance of the GMDH-based model.

Gradient Descent (GD), PSO, and ABC algorithms are contrasted with the outcomes of (Horng *et al*., 2011) introduction of FA to the Radial Basis Function (RBF) Network. The results demonstrated that FA has superior area under curve (AUC) performance in the associated receive operating characteristic (ROC). Mostafaeipour and Qolipour (2018) implemented the FA and Bat Algorithm (BA) to improve the performance of ANN for predicting air travel demand in Iran. The sets of weights used in the neural network model were the results of BA and FA. They reported that the hybrid ANN-FA performed the best as compared to the ANN alone and ANN-BA.

Mahdavi and Zounemat (2019) introduced the integration of GMDH with the harmony search algorithm, GMDH-HS model and GMDH with the firefly algorithm, GMDH-FA model, to estimate air demand on spillway aerators in dams. The GMDH-FA model has superior performance over other models. Mohammadi (2023) proposed the ANN-FA model for drought prediction. The coupled ANN-FA model, which updates and selects the best ANN model weights using the fire-fly algorithm, was introduced to increase the effectiveness of the ANN approach. The ANN-FA is a successful and promising method for predicting drought, according to the results.

By integrating the self-organizing capabilities of GMDH with the global optimization of the Firefly Algorithm (FA), this study proposes a new GMDH-FA model for rainfall forecasting. This combination will make the model significantly more robust and adaptive for capturing complex, nonlinear rainfall dynamics.

II. RELATED WORKS

In order to forecast rainfall, researchers now employ a variety of soft computing methods, such as artificial neural networks (ANN), genetic algorithms (GA), sup-port vector machines (SVR), fuzzy logic and PSO. (Janarthanan *et al*., 2020; Wu and Zhou, 2019; Refonaa, *et al*., 2019; Danandeh Mehr, 2020; Nhita *et al*., 2015; Aksoy & Dahamsheh, 2009). On the other hand, it is possible to predict rainfall using statistical methods as Auto Regressive Moving Average (ARMA), Auto Regressive (AR), Moving Average (MA), ARIMA, and SARIMA (Praveen *et al*., 2020).

An Artificial Neural Network (ANN) is a useful tool for predicting the dynamics of nonlinear systems, and it has been successfully applied in this field for the past 25 years. Nayak *et al*. (2013) provides a comprehensive overview of the literature, covering the multitude of approaches used by different researchers to utilize Arti-ficial Neural Network (ANN) for rainfall prediction. The survey emphasizes that the use of ANN techniques in rainfall prediction outperforms traditional statistical and numerical methods, as demonstrated by the combined results of several stud-ies in this area.

Using rainfall data from Kuching, Kueh and Kuok (2018) employed the Cuckoo Search (CS) Optimization Neural Network model to anticipate rainfall. The Levenberg-Marquardt (LM) optimization technique and the Scale Conjugate Gradient (SCG) were compared. The SCG and LM optimization techniques were surpassed by the CS algorithm. Utilizing Decomposition Genetic Programming (DGP) on several regression equations

Suparta and Samah (2020) employed the Adaptive Neuro Fuzzy Inference System (ANFIS) approach to forecast the amount of rain. The ANFIS model, which has a variety of input structures and membership functions, was developed, trained, and evaluated in order to evaluate a model's potential. Monthly rainfall data analysis spanning six years in South Tangerang City, Banten, showed that rainfall prediction based on ANFIS time series is promising, with 80% of data tests accurately projected.

In predicting rainfall, several ensemble approaches, a range of rainfall data, and cor-responding rainfall variables are used. Chiu *et al*. (2021) examined imputation methods for rainfall forecasting in which the primary characteristics from the me-teorological data are retrieved using principal component analysis (PCA) before imputation. In order to feed the neural network for missing data imputation, the final output of the PCA is combined with the rainfall data from the closest neigh-bouring gauging stations. Next, a technique based on sine and cosine is introduced to enhance neural networks' ability to compensate for the absent rainfall data. The suggested sine cosine function fitting neural network (SC-FITNET) was compared with the sine cosine feedforward neural network (SC-FFNN), feedforward neural network (FFNN), and long short-term memory (LSTM) approaches. The outcomes demonstrated that SC-FITNET performed better than the alternative approaches.

Rahman *et al*. (2022) proposed a unique real-time rainfall prediction system for smart cities based on a fusion strategy that incorporates the best aspects of four popular supervised machine learning approaches: Naïve Bayes, decision trees, K-nearest neighbours, and support vector machines. In order to increase predictive accuracy, the framework combines fuzzy logic with the prediction outcomes of these machine learning techniques. The study uses a dataset of historical Lahore weather data spanning 12 years (2005–2017), pre-processing it as needed by cleaning and normalizing it before moving on to the classification stage. The results show that the machine learning fusion-based framework that was developed is superior to other models, highlighting its potential as a useful tool for real-time rainfall prediction in urban areas.

In the work of Endalie *et al*. (2020), a rainfall forecast model was used to Jimma, a region in southwest Ethiopian Oromia. They made a prediction model that fore-\casts Jimma's daily rainfall using Long Short-Term Memory (LSTM). Experiments were conducted to evaluate the proposed models using measures such as Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Nash–Sutcliffe model efficiency (NSE), and R2. The results were 0.9972, 0.81, 0.4786, and 0.01 in that order. They further compared the proposed model to popular ma-chine-learning regressions, including Multilayer Perceptron (MLP), k-Nearest Neighbors (KNN), Decision Tree (DT), and Support Vector Machine (SVM). The MLP has the lowest RMSE (0.03) of the four learning models that are currently in use. The suggested LSTM model outperforms the existing models.

DemekeEndalie et.al [4] case study for Model of Deep Learning for predicting rainfall. He took the dataset based on agriculture and he gave the results that how the rainfall was going day by day. He proposed the Long-short term memory and gave the correct percentage. Decision tree, K-Nearest Neighbor also used but LSTM has the highest accuracy.

Danandeh *et al*. (2019) presented a novel strategy to develop a hybrid regression model for 1-month-ahead rainfall forecasting at two rain gauge locations (the Tabriz and Urmia stations) in northwest Iran. The method produces accurate rainfall forecasts by integrating the firefly algorithm (FFA) with support vector regression (SVR). Using monthly rainfall data from the gauges that were in the weak station-ary state, the suggested hybrid model was trained and verified. Additionally, the model's efficiency results were cross-validated against those of independent SVR- and genetic programming-based forecasting models that were created as the study's benchmarks. The results demonstrated that the hybrid model performs much better than the benchmarks for both rain gauge locations. The FFA-induced improvement in the SVR forecasts was matched by an almost 30% drop in root-mean-square error and an approximately 100% increase in Nash-Sutcliffe efficiency with regard to the average efficiency results at the gauge locations.

In a comparison study, Barrere-Animas *et al*. (2022) employed simplified rainfall estimation models based on conventional Machine Learning algorithms and Deep Learning architectures that function well for various downstream applications. Comparing models based on LSTM, Stacked-LSTM, Bidirectional-LSTM Networks, XGBoost, and an ensemble of Gradient Boosting Regressor, Linear Support Vector Regression, and an Extra-trees Regressor was necessary to overcome the difficulty of forecasting hourly rainfall volumes using time-series data. Climate data from 2000 to 2020 for five major UK cities were used. The assessment met-rics of Mean Absolute Error, Loss, Root Mean Squared Error, and Root Mean Squared Logarithmic Error were used to evaluate the models' performance. The results indicate that the performance of a Bidirectional-LSTM Network as a rain-fall forecasting model is comparable to that of Stacked-LSTM Networks. The two hidden-layer stacked LSTM network and the bidirectional LSTM network fared the best out of all the models put to the test.

This implies that the models built using LSTM-Networks with fewer hidden layers operate better with this method, suggesting that it can be used as a method for applications involving rainfall forecasting on a budget.

Abdikafi *et al*. (2025) used temperature and humidity data to evaluate three machine learning models namely Long Short -Term Memory (LSTM), Gradient Boost Machine (GBM) and LSTM-GBM. They discovered that the hybrid LSTM-GBM outperformed the other two methods.

Previous research suggested a variety of hybrid models and imputation methods. However, prior research has limited application of the GMDH-based model for rainfall forecasting. Consequently, we would like to examine the rainfall prediction capacity of GMDH-based models. We would employ a number of transfer functions as an alternative to the traditional GMDH model's polynomial function.

III. GROUP METHOD OF DATA HANDLING MODEL

Prof. Ivakhnenko developed a model that only considers input and output relation-ships and does not require human participation (Ivakhnenko, 1971). As a result, the GMDH model is regarded as a data-driven model. According to Tamura and Kondo (1978), Farlow (1981), and Varahrami (2012), Ivakhnenko (1971) pro-posed the fundamental GMDH model, which is particularly helpful in identifying complex as well as unknown nonlinear systems. The GMDH model, like the ANN, has three layers: input, hidden, and output. Every hidden layer that implements a quadratic polynomial's transfer function uses a two-variable combination to con-struct simple neurons. The outputs from this layer will be sent to the neurons in the next layer. However, a threshold value will be used to reduce inefficient neu-rons. The best-performing neurons will be kept, while the worst-performing ones will be removed. Because there is only one neuron in the final layer, the output of this layer will the output of the entire network. The essential idea of the GMDH process is similar to that of a standard neural network built on the forward propagation of impulses via neurons. The capacity of GMDH to spontaneously organize its structures in a heuristic fashion is one of its strongest features (Al-BinHassan & Wang, 2011). With no human intervention, GMDH can generate the neuron number in a layer and the layer number automatically. It also features a self-organizing function that allows it to find the best solution to a problem while avoiding bias and errors (AlBinHassan & Wang, 2011). As a result of this feature, the GMDH has a minimal number of parameters tuned: the largest number of neurons, the highest number of layers, and the pressure of selection, making it a dependable and simple AI model (Ghazanfari *et al*., 2017).

The Kolmogorov-Gabor polynomial (Ivakhnenko, 1971) as shown in the equation below, can be used to simulate the link between variable inputs and outputs in the traditional GMDH.

According to Tamura and Kondo (1978), the comprehensive description in Equation 1 is the most extensively utilised since it can be used to explain practically all real-life systems. The Kolmogorov-Gabor polynomial, which operates as a universal approximator to estimate unknown functions in a system, is important in GMDH modelling (Terasvirta and Kock, 2010).

In each layer, the following second order polynomials of paired variables are joined to form the complete description above.

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk} x_i x_j x_k + \ldots \quad (1)$$

$$y_k = a_0 + a_1 x_i + a_2 x_j + a_3 x_i^2 + a_4 x_j^2 + a_5 x_i x_j \quad (2)$$

The Partial Description (PD) is the above equation, and yk is the intermediate variable. The data is separated into training and testing sets in Step 1. The training data is used to approximate the coefficients, whereas the testing data is needed to pick intermediate variables. Step 2 follows: The parameters a_0, a_1, a_2, a_3, a_4, and a_5 in Equation 2 are calculated using the least square approach utilising training data based on the combination of two input variables x_i and x_j. Step 3: Using the PD estimated in Step 2, calculate the regularity criterion for the checking data. MSE is the regularity criterion in classic GMDH.

The variables in the middle that have the minimum MSE will be chosen as helpful variables. Step 4: Replace variables x_i and x_j with y_i and y_j, respectively. Steps 2–5 are performed until the variables' performance cannot be improved any further. The basic design process of GMDH neuron structure is illustrated in Fig. 1. A linear polynomial is employed as the transfer function in the traditional GMDH.

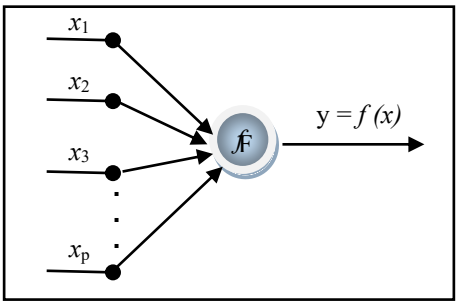

Fig. 1. GMDH neuron structure

IV. FIREFLY ALGORITHM

The Firefly Algorithm (FA), introduced by Yang (2010), is employed in this study as an optimization technique to determine the optimal weights for combining multiple GMDH models. In the proposed framework, each firefly represents a candidate solution in the form of a weight vector assigned to the outputs of different GMDH variants, namely GMDH-Polynomial, GMDH-Sigmoid, GMDH-Tangent, and GMDH-RBF. The quality of each solution is evaluated using an objective function, typically defined based on forecasting error measures such as the Sum of Squared Errors (SSE), Root Mean Square Error (RMSE), or Mean Absolute Error (MAE).

In FA, the brightness of a firefly corresponds to the fitness value of the solution, where brighter fireflies indicate better-performing weight combinations. Fireflies are assumed to be unisex, and their attractiveness is proportional to their brightness while decreasing exponentially with distance. Accordingly, a firefly with lower brightness moves toward a brighter one based on the attractiveness function $\beta(r)=\beta_0 e^{-\gamma r^2}$, where β_0 is the attractiveness at zero distance and γ is the light absorption coefficient. The movement of a firefly is governed by both deterministic attraction toward better solutions and a randomization component, enabling a balance between global exploration and local exploitation.

During each iteration, the positions of fireflies that is weight vectors are updated, and their corresponding fitness values are recalculated. If a firefly does not find a brighter counterpart, it moves randomly within the search space to maintain diversity. This iterative process continues until a predefined stopping criterion, such as the maximum number of iterations, is reached. The best firefly obtained represents the optimal set of weights, which is then used to combine the outputs of the individual GMDH models. Fig. 2 shows the FA pseudocode.

The Firefly Algorithm (FA) begins by defining an objective function f(x), where $x=(x_1,\ldots x_d)^T$ represents a solution in a d-dimensional search space. An initial population of N fireflies is then generated, with each firefly positioned at $x_a=[w_a]$, for a=1,2,…,N. The light intensity I_a of each firefly is evaluated using the objective function $f(x_a)$, and a light absorption coefficient γ is specified to control the reduction of attractiveness with distance. The algorithm proceeds iteratively while the generation counter t is less than the maximum number of generations (HighestGeneration). Within each iteration, every firefly a is compared with all other fireflies b (a=1,2,…,N,b=1,2,…,N); if $I_a < I_b$, firefly a moves toward firefly b. The movement is governed by an attractiveness function that decreases with distance r, typically modeled using $\exp(-\gamma r)$. After updating positions, the new solutions are evaluated and the light intensities I_a are updated accordingly. Once all fireflies have been processed, they are ranked based on their intensities, and the current global best solution g* is identified. This process repeats until the stopping criterion is met, after which the final solution is obtained through post-processing.

By integrating FA into the GMDH framework, the proposed GMDH-FA model effectively optimizes the contribution of each transfer function, thereby enhancing forecasting accuracy and improving generalization performance for rainfall prediction tasks.

```
1   Objective function f(x); x = (x1, … , xd)T
2   Produce early populace of fireflies places xa = [wa] (for a = 1, 2, …
    , N)
3   Light intensity Ia at xa is defined by f(xa)
4   State light absorption coefficient γ
5   while (t < HighestGeneration)
6   for a = 1 : N all N fireflies
7       for b = 1 : N all N fireflies (inner loop)
8         if (Ia < Ib ), Shift firefly a to b; end if
9               Differ attraction with distance r via exp[-γ r]
10              Analyze current resolutions and renew light intensity
11      end for b
12  end for a
13  Rate the fireflies and recover the latest global best g*
14  end while
15  Results postprocess and conception
```

Fig. 2. FA Pseudocode

V. METHODS

A. Dataset

This study used monthly rainfall information for Kuching from 2010 to 2019. The nearby meteorological stations are used to collect rainfall data. The monthly rainfall series is displayed in Fig. 3. The objective of the is to forecast the city of Kuching's monthly rainfall. The data was received from Malaysia's Department of Meteorology are unavoidable.

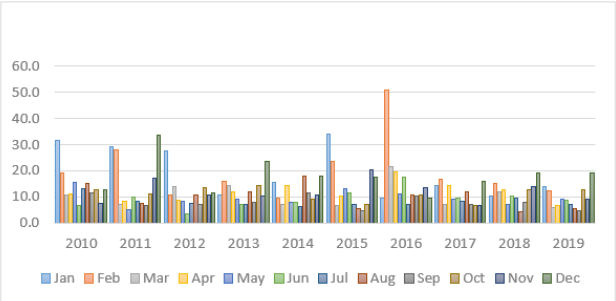

Fig. 3. Kuching Monthly Rainfall (2010 – 2019)

Before entering a forecasting model, a dataset must go through the pre-processing data stage. The steps involved in this study, which intends to conduct time series forecasting, are data transformation, data division into training and testing sets, lagging the data, and selecting appropriate inputs.

The data is rescaled into a smaller range, such as 0 to 1, -1 to 1, and so on, for data transformation. To prevent computing enormous numbers, smaller numbers are required. The data is standardized into a range of 0 to 1 in this study using the following formula [28]:

$$Q^1 = \frac{Q - s}{r - s} \tag{3}$$

where Q1 is the normalized data, Q is the actual data, r and s are the maximum and minimum values of the actual data, respectively.

When forecasting employs 10 years of daily rainfall data to predict daily rainfall, massive calculation issues may arise. We used monthly rainfall averages as an alternative. Training and testing sets are created from the data. Additionally, divide the data into training and testing sets, with the training sets being used to construct the model and the testing sets being used to test the correctness of the model. The model is tested by making predictions against the testing sets once it has been constructed using the training sets. If all the data is used for model creation as well as forecasting, the model may become overfit. The chosen meteorological characteristics will be trained and tested to forecast rainfall in this project. The training data size is 90%, whereas the testing data size is 10%.

B. Experimental Setup

The first experiment in this study involves placing polynomial, sigmoid, tangent and radial basis functions for the GMDH model individually. The forecasts from these individual models were then combined to produce combined forecasts. In the second experiment, the combined forecasts from these models become the input to the FA model. Fig. 4 illustrates the proposed GMDH-FA model in this study.

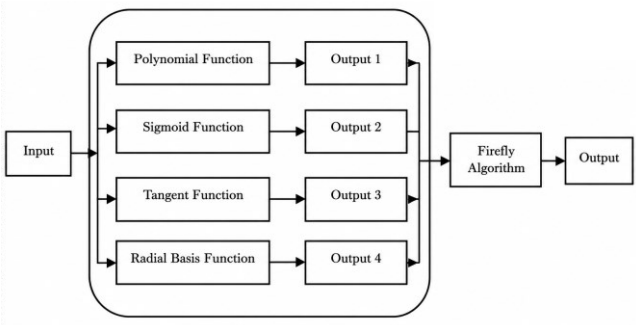

Fig. 4. The Proposed Combined GMDH-FA Model

C. Models of Development for Independent GMDH

A linear polynomial is employed as the transfer function in the traditional GMDH. Therefore, the term "GMDH-Poly" will be used in this study going forward to refer to the ordinary GMDH. The basic design process of GMDH-Poly is illustrated in Fig. 1. In the first step, (a) the definition of the time series input variables is $X = \{x_1, x_2, ..., x_P\}$ is made. The training and testing sets are the first divisions made of the data. During the learning phase, when the GMDH network is created, the training set will be applied. In the meantime, the model's performance will be evaluated using the testing set. Next step, (b) using the training data, inputs are entered into the GMDH model two at a time. The following formula is used to determine how many combinations there are in each layer:

$$L = \frac{M(M-1)}{2} \tag{4}$$

Each combination of input in a traditional GMDH will create a PD using a polynomial, as the example below illustrates:

$$z = a_0 + bx_1 + cx_i + dx_1x_2 + ex_1^2 + fx_2^2 \tag{5}$$

Regression is used in the above equation to determine the coefficients a, b, c, d, e, and f.

The neuron input, yk, in the linear polynomial transfer function is computed as follows:

$$y_k = z_k \tag{6}$$

Then, in step € each neuron's performance will be determined using RMSE once the coefficients for every neuron in a layer have been determined. To pick high-performing neurons in each layer, a single parameter needs to be set as the appropriate threshold in this step. The average of the lowest and highest RMSE generated by the layer's neurons serves as the threshold value in this investigation. The neurons that go below this threshold will be eliminated. In the next step (d), each layer repeats steps (a) through (c) until the halting requirement is met. In this study, the procedure will come to an end when the allotted maximum layer value is attained. It should be noticed that the last layer only chooses one neuron. Then in step €, the testing data will be entered into the GMDH structure when it has been created in order to provide a forecast. Lastly, in step (f), repeating the steps (a) through € using three more non-polynomial transfer functions (Logistic Sigmoid Function, Radial Basis Function and Hyperbolic Tangent Function) in place of the polynomial transfer function from step (b). The following Table I shows the equation for each transfer function used in this study.

TABLE I. TRANSFER FUNCTIONS

Transfer Function	Equation	
Hyperbolic Tangent	$y_{tan} = \frac{e^{2z}-1}{e^{2z}+1}$	(7)
Sigmoid	$y_{sig} = \frac{1}{1+e^{-z_k}}$	(8)
Radial Basis Function	$y_{rbf} = e^{-z_2^k}$	(9)

And the value z_k for every transfer function be approximated via the following formulation;

1. Logistic sigmoid function

$$z_k = log\left(\frac{\emptyset'}{1-\emptyset'}\right) \quad (10)$$

2. Hyperbolic tangent function

$$z_k = \frac{1}{2} log\left(\frac{1+\emptyset'}{1-\emptyset'}\right) \quad (11)$$

3. RBF

$$z_k = \sqrt{-log\emptyset'} \quad (12)$$

Where, ϕ' : the normalized output variable (ranging from zero to one), ϕ : the output variable.

D. Weight generation with the Firefly Algorithm

The first model, GMDH-Poly will be forecasted as y_1 in this work, followed by the second, GMDH-Sigmoid as y_2, the third, GMDH-Tangent as y_3, and the last GMDH-RBF as y_4. The proposed model's process has five steps. First, (a) FA receives the forecasts from each individual model. Next, (b) FA will first produce a set of weights at random. The four models will be subjected to these weights in order to generate a new forecast, as indicated by the following equation:

$$\hat{y} = \sum_{k=1}^{N} w_k \, y_k \quad (13)$$

where N is the total number of unique models employed, wk is the weights, and ŷ is the new forecast value. Further, © sum of squared errors (SSE) will be used to determine the accuracy of the new output, as indicated below:

$$SSE = \sum_{i=1}^{n} (y - \hat{y})^2 \quad (14)$$

where y is either the actual value or the original series value, and n is the number of observations. Then, (d) FA generates fresh random weights in this stage. Step 3 involves determining the SSE of the new output after a new combination model is created using Equation (10) in a manner similar to Step 2. The former output and this new output will be contrasted. The weights that yield the lowest SSE value will be retained, swapping out the underperforming weights. Lastly, (e) repeat these procedures until the maximum number of iterations is reached. The final new forecast will be created by combining the four models using the best weights that were determined at the conclusion of the iterations.

VI. THE EXPERIMENTAL RESULTS

In order to assess the dispersion between observed and predicted data, performance measurement is essential. It is also utilized to accurately analyze the model's reliability and strength in forecasting. Furthermore, performance metrics make it easy to make comparisons between different forecasting models. The well-known measurement Root-Mean-Square Error (RMSE), the Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), are frequently employed in time series forecasting [1][19][31].

The RMSE, MAE and MAPE are used to evaluate and select each model's performance for both training and test data. Therefore, the RMSE, MAE and MAPE are used to analyse the results of rainfall forecasting using the GMDH-based model. The following formulae are used to calculate the RMSE, MAE and MAPE respectively.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \quad (15)$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \quad (16)$$

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \quad (17)$$

Where y_t represents the original value, $\hat{y}_t$ represents the predicted value, and n is the sample size. During the evaluation phase, a distribution is said to be more suited to the actual data if the values for its RMSE, MAE, and MAPE are smaller. The suggested models' performances will be compared to the benchmark models using these three values, and the model that generates the lowest RMSE, MAE, and MAPE is deemed to be the best model. These three metrics are used to compare the GMDH models' performances. The conventional GMDH model using polynomial transfer function is referred to as GMDH-Poly. Tables II and III present the performance comparison of the conventional GMDH model and the GMDH-FA model where the best performances are in bold. To assess the dispersion between

observed and predicted data, performance measurement is essential. It is also utilized to accurately analyze the model's reliability and strength in forecasting.

TABLE II. PERFORMANCE RESULTS FOR THE TRAINING SET

Model	RMSE	MAE	MAPE
GMDH-Poly	0.10303	0.06730	26.43907
GMDH-Sigmoid	0.0991	**0.0618**	23.3521
GMDH-Tan	0.1020	0.0680	27.2258
GMDH-RBF	0.1002	**0.0618**	**23.3429**
GMDH-FA	**0.0966**	0.0645	25.6869

TABLE III. PERFORMANCE RESULTS FOR THE TESTING SET

Model	RMSE	MAE	MAPE
GMDH-Poly	0.09810	0.07607	47.38644
GMDH-Sigmoid	0.0844	0.0665	41.3093
GMDH-Tan	0.1036	0.0760	48.1440
GMDH-RBF	0.0821	0.0657	40.7722
GMDH-FA	**0.0550**	**0.0455**	**24.0733**

Tables II and III show that all GMDH-based models have comparable results on the training data, while the hybrid GMDH-FA model performs best on the testing dataset. In training, GMDH-FA had the lowest RMSE, although GMDH-Sigmoid and GMDH-RBF showed slightly better MAE and MAPE values, respectively. In testing, GMDH-FA performs better than the other models in RMSE (0.0550), MAE (0.0455), and MAPE (24.0733), confirming that the model is performing well and has strong predictive capability. Conventional GMDH models, such as GMDH-Polynomial and GMDH-Tan, provide relatively higher error values and show a significant degradation in performance from training to testing, indicating overfitting. In contrast, the improved performance on unseen data by GMDH-FA highlights the effectiveness of the Firefly Algorithm for the optimization of model weights and generalization. In summary, the results support that the hybrid GMDH-FA model can serve as a more accurate and reliable approach for rainfall forecasting than individual GMDH variants.

VII. CONCLUSION

Rainfall forecasting remains a challenging and essential research area due to the nonlinear, dynamic, and uncertain nature of meteorological data. This study proposed a hybrid intelligent model that integrates the Group Method of Data Handling (GMDH) with the Firefly Algorithm (FA) to enhance the accuracy and robustness of rainfall predictions. The proposed framework exploits the self-organizing and nonlinear modeling capability of GMDH, while utilizing the global optimization strength of FA to refine model weights and structural parameters. The superior performance of the GMDH-FA model over the conventional GMDH-Poly demonstrates the effectiveness of FA in improving model optimization and enhancing the ability to capture complex rainfall patterns.

Overall, the findings indicate that the GMDH-FA model provides improved predictive accuracy and better generalization compared to traditional GMDH approaches, making it a promising tool for rainfall forecasting. The successful application of this hybrid model suggests its potential extension to other hydrological forecasting tasks, where accurate predictions are critical for effective planning, resource management, and disaster risk reduction.

Empirical results further reveal that although GMDH-based models exhibit comparable performance during the training phase, the hybrid GMDH-FA consistently achieves superior results on the testing dataset. Specifically, it records the lowest RMSE, MAE, and MAPE values, confirming its strong predictive capability and robustness. In contrast, conventional GMDH variants, such as GMDH-Polynomial and GMDH-Tangent, demonstrate higher error values and notable performance degradation from training to testing, indicating overfitting. The improved performance of GMDH-FA on unseen data highlights the contribution of FA in enhancing model generalization through effective weight optimization.

In conclusion, the hybrid GMDH-FA model offers a more accurate and reliable alternative to individual GMDH models for rainfall forecasting. Future research should focus on validating the model across diverse climatic regions to assess its scalability and generalization capability, as well as incorporating cross-validation techniques to improve model stability. Additionally, extending the framework to long-term rainfall forecasting could further support applications in agriculture and water resource management.

ACKNOWLEDGMENT

We especially appreciate the rainfall data provided by the Malaysian Meteorological Department. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] Ivakhnenko, A. G. (1971). Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, *4*, 364–378.

[2] Tamura, H., & Kondo, T. (1978). Revised GMDH algorithm using prediction sum of squares (PSS) as a criterion for model selection. *Transactions of the Society of Instrument and Control Engineers*, *14*(5), 519–524.

[3] Tauser, J., & Buryan, P. (2011). Exchange rate predictions in international financial management by enhanced GMDH algorithm. *Prague Economic Papers*, *20*(3), 232–249.

[4] Ugrasen, G., Ravindra, H. V., Prakash, G. N., & Keshavamurthy, R. (2014). Estimation of machining performances using MRA, GMDH and artificial neural

network in wire EDM of EN-31. *Procedia Materials Science*, *6*, 1788–1797.

[5] Danandeh Mehr, A. (2020). Seasonal rainfall hindcasting using ensemble multi-stage genetic programming. *Theoretical and Applied Climatology*. https://doi.org/10.1007/s00704-020-03438-3.

[6] Janarthanan, R., Balamurali, R., Annapoorani, A., & Vimala, V. (2021). Prediction of rainfall using fuzzy logic. *Materials Today: Proceedings*, *37*, 959–963.

[7] Wu, J., & Zhou, J. (2009). Support vector regression based on particle swarm optimization and projection pursuit technology for rainfall forecasting. In *2009 International Conference on Computational Intelligence and Security*.

[8] Refonaa, J., Lakshmi, M., Srinivasa Rao, R. S. S., & Prasad, E. (2019). Rainfall prediction using genetic algorithm. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*(2S3).

[9] Onwubolu, G. C., Buryan, P., Garimella, S., Ramachandran, V., Buadromo, V., & Abraham, A. (2007). Self-organizing data mining for weather forecasting. In *IADIS European Conference on Data Mining*.

[10] Jirina, M. (1994). The modified GMDH sigmoidal and polynomial neural net. *IFAC Proceedings*, *27*(8), 611–613.

[11] Kondo, T., & Pandya, A. S. (2003). Structural identification of the multi-layered neural networks by using revised GMDH-type neural network algorithm with a feedback loop. In *SICE 2003 Annual Conference* (Vol. *3*, pp. 2768–2773). IEEE.

[12] Narawi, A., Jawawi, D. N. A., & Samsudin, R. (2022). Rainfall forecasting using the group method of data handling model: A case study of Sarawak, Malaysia. In *Lecture Notes on Data Engineering and Communications Technologies* (Vol. *127*, pp. 129–140). Springer.

[13] Nourani, V., & Komasi, M. (2013). A geomorphology-based ANFIS model for multi-station modeling of rainfall-runoff process. *Journal of Hydrology*, *490*, 41–55.

[14] Yang, X. S. (2008). *Nature-inspired metaheuristic algorithms*. Luniver Press.

[15] Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimization. *International Journal of Bio-Inspired Computation*, *2*(2), 78–84.

[16] Yang, X. S., & He, X. S. (2013). Firefly algorithm: Recent advances and applications. *International Journal of Swarm Intelligence*, *1*(1), 36–50.

[17] Horng, M. H., Lee, Y. X., Lee, M. C., & Liou, R. J. (2011). Firefly meta-heuristic algorithm for training the radial basis function network for data classification and disease diagnosis. In *Theory and New Applications of Swarm Intelligence* (pp. 115–132).

[18] Xiao, L., Shao, W., Liang, T., & Wang, C. (2016). A combined model based on multiple seasonal patterns and modified firefly algorithm for electrical load forecasting. *Applied Energy*, *167*, 135–153. http://dx.doi.org/10.1016/j.apenergy.2016.01.050.

[19] Mostafaeipour, A., Gol, A., & Qolipour, M. (2018). Prediction of air travel demand using a hybrid artificial neural network (ANN) with bat and firefly algorithms: A case study. *The Journal of Supercomputing*. https://doi.org/10.1007/s11227-018-2452-0.

[20] Mahdavi-M., A., & Zounemat-K., M. (2019). A new integrated model of the group method of data handling and the firefly algorithm (GMDH-FA): Application to aeration modelling on spillways. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-019-09741-4.

[21] Mohammadi, B. (2023). Modeling various drought time scales via a merged artificial neural network with a firefly algorithm. *Hydrology*, *10*, 58. https://doi.org/10.3390/hydrology10030058.

[22] Danandeh Mehr, A. (2020). Seasonal rainfall hindcasting using ensemble multi-stage genetic programming. *Theoretical and Applied Climatology*. https://doi.org/10.1007/s00704-020-03438-3.

[23] Nitha, F., Adiwijaya, Annisa, S., & Kinasih, S. (2015). Comparative study of grammatical evolution and adaptive neuro-fuzzy inference system on rainfall forecasting in Bandung. In *2015 3rd International Conference on Information and Communication Technology (ICoICT)*. IEEE.

[24] Aksoy, H., & Dahamsheh, A. (2009). Artificial neural network models for forecasting monthly precipitation in Jordan. *Stochastic Environmental Research and Risk Assessment*, *23*, 917–931. https://doi.org/10.1007/s00477-008-0267-x.

[25] Praveen, B., Talukdar, S., Shahfahad, Mahato, S., Mondal, J., Sharma, P., Islam, A. B. M. T., & Rahman, A. (2020). Analyzing trend and forecasting rainfall changes in India using nonparametrical and machine learning approaches. *Scientific Reports*, *10*, 10342. https://doi.org/10.1038/s41598-020-67228-7.

[26] Nayak, D. R., Mahapatra, A., & Mishra, P. (2013). A survey on rainfall prediction using artificial neural network. *International Journal of Computer Applications*, *72*(16).

[27] Danandeh Mehr, A., Nourani, V., Karimi Khosrowshahi, V., & Ghorbani, M. A. (2019). A hybrid support vector regression-firefly model for monthly rainfall forecasting. *International Journal of Environmental Science and Technology*, *16*(1), 335–346.

[28] Kueh, S. M., & Kuok, K. K. (2018). Forecasting long term precipitation using cuckoo search optimization neural network models. *Environmental Engineering and Management Journal*, *17*(6), 1283–1291.

[29] Suparta, W., & Samah, A. A. (2020). Rainfall prediction by using ANFIS time series technique in South Tangerang, Indonesia. *Geodesy and Geodynamics*, *11*, 411–417.

[30] Chiu, P. C., Selamat, A., Krejcar, O., Kuok, K. K., Herrera-Viedma, E., & Fenza, G. (2021). Imputation of rainfall data using the sine cosine function fitting neural network. *International Journal of Interactive Multimedia and Artificial Intelligence*. http://dx.doi.org/10.9781/ijimai.2021.08.013.

[31] Rahman, A. U., Abbas, S., Gollapalli, M., Ahmed, R., Aftab, S., Ahmad, M., & Mosavi, A. (2022). Rainfall prediction using multiple linear regression model. In *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*. IEEE.

[32] Endalie, D., Haile, G., & Taye, W. (2022). Deep learning model for daily rainfall prediction: Case study of Jimma, Ethiopia. *Water Supply*, *22*(3), 3448–3461.

[33] Barrera-Animas, A. Y., Oyedele, L. O., Bilal, M., Akinosho, T. D., Delgado, J. M. D., & Akanbi, L. A. (2022). Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. *Machine Learning with Applications*, *7*, 100204.

[34] Ahmed, H. A. Y., & Mohamed, S. W. A. (2021). Rainfall prediction using multiple linear regression model. In *2020*

International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE). IEEE.

[35] Agrawal, A., Adke, A., Hood, V., Bambale, R., & Shelke, P. (2024). Forecasting rainfall utilizing simple linear regression. In *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)* (pp. 1348–1352). IEEE.

[36] AlBinHassan, N. M., & Wang, Y. (2011). Porosity prediction using the group method of data handling. *Geophysics*, *76*(5), O15–O22.

[37] Ghazanfari, N., Gholami, S., Emad, A., & Shekarchi, M. (2017). Evaluation of GMDH and MLP networks for prediction of compressive strength and workability of concrete. *Bulletin de la Société Royale des Sciences de Liège*, *86*(Special Edition), 855–868.

[38] Kock, A. B., & Teräsvirta, T. (2010). Forecasting with nonlinear time series models. *CREATES Research Paper*, *2010*(1), 1–31.

[39] Shabri, A., & Samsudin, R. (2014). A hybrid GMDH and Box-Jenkins model in time series forecasting. *Applied Mathematical Sciences*, *8*(62), 3051–3062.

[40] Danandeh Mehr, A. (2020). An ensemble genetic programming model for seasonal precipitation forecasting. *SN Applied Sciences*, *2*(11), 18210.

[41] Danandeh Mehr, A., Nourani, V., Karimi Khosrowshahi, V., & Ghorbani, M. A. (2019). A hybrid support vector regression-firefly model for monthly rainfall forecasting. *International Journal of Environmental Science and Technology*, *16*, 335–346.

[42] Danandeh Mehr, A. (2021). Seasonal rainfall hindcasting using ensemble multi-stage genetic programming. *Theoretical and Applied Climatology*, *143*(1–2), 461–472.

[43] Abdikafi, E. A., Abdihamid, M. T., & Abdisalam, A. A. (2025). Hybrid machine learning model for rainfall prediction using time-series data. *International Journal of Engineering Trends and Technology*, *73*(6), 17–29.