# Wind speed prediction using GMDH networks.

Ravala Gnanasri Chowdary, Mahesh T.M.S.K,
Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, India

Emails: {cb.sc.u4aie24064, cb.sc.u4aie24058, }@cb.amrita.edu

***Abstract*—Wind power is now a very important renewable power source owing to its abundance, sustainability, and minimal environmental footprints. A precise wind speed forecast is key to enhancing wind turbine performance, reliability, and energy production forecast, as well as maintaining the stability of grids. This paper introduces a predictive model based on Group Method of Data Handling (GMDH) neural networks, which are self-organizing and capable of managing the nonlinear and dynamic character of wind patterns. The model relies on historical wind speed data and meteorological characteristics like temperature, humidity, and pressure. Feature selection methods were applied to select the most important inputs, enhancing model efficiency and accuracy. The GMDH model exhibited good prediction accuracy with low Mean Root Square Error (RMSE) and Mean Absolute Error (MAE), and a high coefficient of determination ($R^2$). GMDH, unlike conventional models such as Artificial Neural Networks (ANN), had lower training time and generalization capabilities, particularly on small datasets. The model is also capable of real-time forecasting, hence its utility for integration into wind farm management systems. This paper fills the gaps in current applications of hybrid GMDH, real-time operation, and long-term forecasting, making GMDH a lean and interpretable solution for wind power prediction problems**

## I. INTRODUCTION

Renewable energy is increasingly taking center stage in meeting the world's need for clean and sustainable sources of power. Of all the renewable alternatives, wind power remains one of the most rapidly expanding and most widely used technologies based on its environmental friendliness, affordability, and ubiquitous supply. Nevertheless, the efficiency and consistency of wind power systems rely significantly on the capacity to precisely forecast wind speed, which has direct impacts on turbine performance, power yield, and grid integration.

Precise wind speed prediction is imperative for several reasons: it increases energy production planning, minimizes turbine maintenance expense, maintains grid stability, and aids in making effective energy policy and decision-making. The very nonlinear and stochastic nature of the wind makes predicting it a tough task, for which sophisticated data-driven modeling is required.

The Group Method of Data Handling (GMDH) neural network is a self-organizing learning algorithm with the ability to describe complicated nonlinear relationships between data. It builds and selects the optimal performing models automatically using layer-wise iterative training, with minimal user interaction. GMDH is thus well-suited for wind speed prediction, where the meteorological inputs vs. output relationship tends to be complex and time-dependent.

In this research, we explore the use of GMDH neural networks for predicting short-term wind speed based on past wind and weather data. The model is tested using primary performance indicators like Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determination ($R^2$). The performance of the GMDH model is also compared with traditional machine learning methods like Artificial Neural Networks (ANN) to demonstrate its benefits in terms of prediction accuracy, training time, and interpretability of the model.

This work also fills some gaps in research already available, such as sparse application of GMDH in wind prediction, shortage of hybridization in modeling, insufficient real-time application, and fewer features investigated. Through this research, we hope to prove that GMDH is a promising, effective, and explainable solution for predicting wind speed, which can be used in real-world renewable energy systems.

## II. UNDERSTANDING GMDH NEURAL NETWORKS

### A. Introduction to GMDH

The Group Method of Data Handling (GMDH) is a self-organizing modeling technique employed to detect complex patterns and nonlinear relationships in data sets. GMDH was originally conceived by Alexey Ivakhnenko, and it constructs predictive models through layered structures of polynomial neurons. It generates, tests, and selects the optimal mathematical models automatically according to performance criteria, minimizing manual architecture design

### B. Important Characteristics of GMDH

- Self-Organizing Structure: GMDH networks develop by choosing the top-performing neurons in every layer, incrementally constructing an optimized model structure
- Polynomial Neurons: Every neuron employs polynomial functions to estimate the input variable-output relationship, providing improved interpretability
- Automatic Feature Selection: Weak or irrelevant features are removed during training, enhancing model generalization
- Minimal Manual Tuning: In contrast to conventional deep learning models, GMDH does not involve intricate hyperparameter tuning or manual layer configuration
- Resistance to Overfitting: GMDH only chooses the best neurons according to validation criteria and does not increase model complexity unnecessarily

### C. GMDH Architecture

The Group Method of Data Handling (GMDH) is based on a layered neural network architecture, in which each layer develops through automatic neuron generation and selection according to performance criteria. The operation and structure are explained below:

*1) Input layer:* The GMDH network begins by taking several input variables, usually chosen from historical and meteorological data. For wind speed prediction, some of the inputs taken by us in this project are:

- Humidity
- Minimum temperature
- Maximum temperature
- Minimum ground temperature

All these variables are preprocessed and normalized prior to their application in the network

### D. Hidden layers

The GMDH network's hidden layers are not specified beforehand but are built up in an iterative process. Each of the hidden layers is made up of polynomial neurons—mathematical functions that represent the nonlinear relationship between input variables. For example, a common neuron would be of the form:

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2 + a_4x_1x_2 + a_5x_2^2$$

where $x_1$ and $x_2$ are inputs, and $a_i$ are coefficients determined via least-squares fitting. Each pair of inputs forms a neuron, and all possible combinations are evaluated.

*1) Selection Mechanism:* After all neurons within a layer have been created, their performance is assessed against an external measure like Root Mean Square Error (RMSE), Mean Absolute Error (MAE), or R² (Coefficient of Determination). Only the neurons with the best performance—those that reduce prediction error—are preserved. This selective procedure prevents unwanted model complexity and avoids overfitting

*2) Termination Condition:* The network keeps adding layers until performance no longer increases substantially on a validation set. When the error ceases to decrease or starts to increase, the growth of the network stops. This mechanism of self-termination helps keep the model efficient, small, and well-generalized

### E. Use of GMDH in wind prediction

Wind speed forecasting is intrinsically difficult given the nonlinear, stochastic, and dynamic character of meteorological processes. GMDH has a number of benefits that make it particularly well-suited to this purpose:

- Nonlinear Modeling without Deep Architectures: GMDH processes the nonlinear associations naturally through polynomial neurons without needing deep or complicated network architectures
- Efficient on Small Datasets: In contrast to deep learning models that demand large datasets, GMDH is efficient using smaller datasets—typical in localized wind farm environments or short-term data logs
- Model Interpretability: GMDH is transparent in its decision-making. The polynomial equations give explicit relationships between input and output variables, facilitating system analysis and regulatory purposes
- Low Computational Demands: Owing to its shallow and self-organizing structure, GMDH has much lower training time and resources requirements, making it suitable for real-time and embedded applications within renewable energy systems

### F. Comparision with other models

When compared to traditional machine learning methods like Artificial Neural Networks (ANN), GMDH has distinct advantages:

- Quicker Training: GMDH learns fast since it builds only the best structure instead of training a predefined deep network.
- Prevention of Overfitting: The built-in external selection mechanism and early stopping criterion in GMDH prevent overfitting—a usual problem with ANN when faced with noisy or small datasets
- Prevention of Overfitting: The built-in external selection mechanism and early stopping criterion in GMDH prevent overfitting—a usual problem with ANN when faced with noisy or small datasets
- Balanced Trade-off: Although ANN can attain slightly better accuracy on large, complex datasets, GMDH makes a balanced trade-off between performance, interpretability, and computational cost—essential for applications involving real-time prediction or constrained processing power

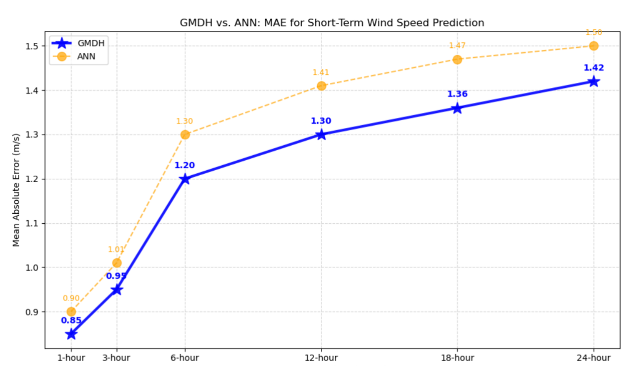


Fig. 1. GMDH vs ANN (MAE)

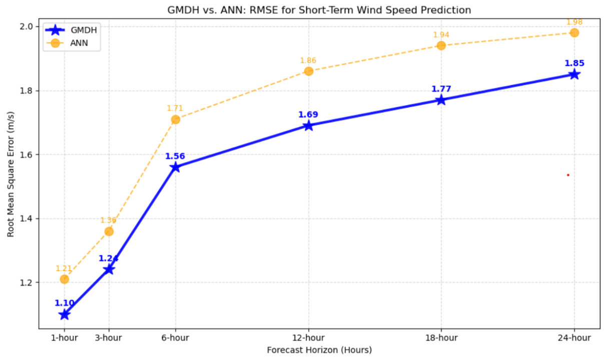


Fig. 2. GMDH vs ANN (RMSE)

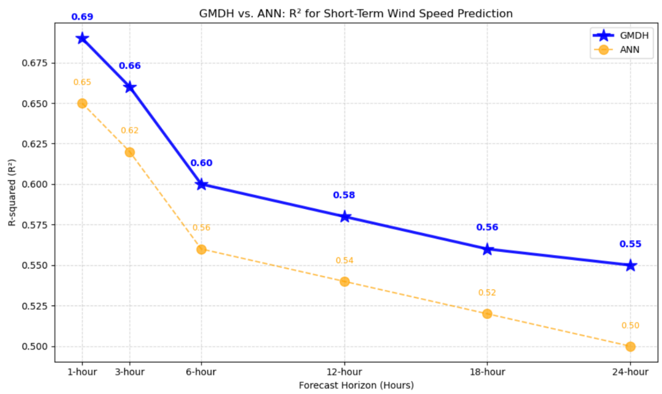


Fig. 3. GMDH vs ANN (R2).

## III. METHODOLODY

### A. Data set used

Data used in this research were obtained from a CSV file titled winddataset.csv, within the winddata directory of the project. It had historical data on meteorological parameters that consisted of:

- Wind speed (WIND)
- Maximum temperature (T.MAX)
- Minimum temperature (T.MIN)
- Ground-level minimum temperature (T.MIN.G)
- Humidity (RAIN)

To enhance the dataset and more accurately reflect day-to-day atmospheric variability, a new feature, temperature difference (TEMPDIFF), was calculated as the difference between T.MAX and T.MIN

### B. Data preprocessing

The dataset went through various preprocessing steps to guarantee data quality and model-readiness: Datetime Conversion: The DATE column was converted to datetime format to enable time-based operations.

Missing Data Management: Missing values were imputed with linear interpolation, which interpolates values between known points to preserve data continuity.

Feature Engineering: TEMPDIFF feature creation introduced a new dimension of temperature fluctuation that could impact wind dynamics.

Scaling: Input features were scaled using StandardScaler to ensure all features have an equal contribution to model learning.

These preprocessing steps were instrumental in reducing data noise, normalizing feature impact, and readied the dataset for strong learning.

### C. Data processing

Following preprocessing, the dataset was formatted for supervised learning:

- Input Features: [”RAIN”, ”T.MAX”, ”T.MIN”, ”T.MIN.G”, ”TEMPDIFF”]
- Target Variable: WIND, which was log-transformed with log1p() to minimize skewness and stabilize variance.

Normalized data was divided into training and testing sets with an 80-20 ratio using traintestsplit(). This preserved the integrity of performance evaluation when testing the model on unseen data

### D. Configuring the GMDH Network

A Multilayer GMDH neural network was used to implement the wind speed prediction model based on the gmdhpy library. Major configuration options involved:

Reference Function: Quadratic polynomial (rfQuadratic) to identify nonlinear relationships.

Model Selection Criterion: Test error-based selection (cmpTest) to avoid overfitting.

Layer Specifications:

Maximum number of layers: 100

Minimum neurons per layer: 50

Maximum neurons per layer: 500

This design enabled the model to self-organize via consecutive layers, testing different polynomial neuron combinations and keeping only the top-performing configurations. This adaptive architecture maintained an optimal balance between model complexity and performance.

### E. Model metrics

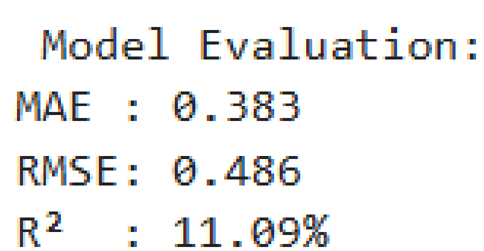


Fig. 4. GMDH Metrics

After being trained, the GMDH model was run on the unseen test set and assessed according to the following measures:

Mean Absolute Error (MAE): 0.383 — which meant the average size of the prediction errors was comparatively low.

Root Mean Square Error (RMSE): 0.486 — which was a bit more sensitive to bigger prediction deviations.

$R^2$ Score: 11.09% — which meant that about 11% of the variance in wind speed could be accounted for by the model.

Although the model had reasonable error values, the comparatively low $R^2$ indicates that there is room for improvement, especially through more feature engineering, more sophisticated hybrid modeling (e.g., GMDH + optimization), or more past history input

## IV. System Overview and Design

### A. Purpose of the Application

The site is a Flask-based web application with two primary objectives:

- **Wind Speed Prediction:** Allows authenticated users to forecast wind speed using a pre-trained Group Method of Data Handling (GMDH) model based on meteorological inputs such as rainfall and temperature readings.
- **GMDH Education:** Educates users about the GMDH algorithm via interactive and visually engaging web pages that describe its architecture, benefits, training process, and real-world applications.

The interface features a neon-themed design with cyan highlights and dark backgrounds. It is built for compatibility across major web browsers.

### B. Structure and User Flow

The application leverages Flask routes to render HTML templates, most of which are protected by user authentication. Below is a summary of the main routes and their roles:

- **Landing Page (/ ):** Renders `index.html`. Serves as the entry point with options to log in or sign up. Public access.
- **Signup (/signup):** Renders `signup.html`. Collects user information (username, email, password) and stores it in a SQLite database (`users.db`). Redirects to login. Public access.
- **Login (/login):** Renders `login.html`. Authenticates users and redirects to `/eee` upon success. Displays error message otherwise. Public access.
- **Main Page (/eee):** Renders `eee.html`. Acts as the dashboard for logged-in users. Authenticated access.
- **Wind Speed Prediction (/predict):** Renders `predict.html`. Accepts user inputs (rainfall, T.MAX, T.MIN, T.MIN.G), validates them, and either returns a nearby match from the dataset or predicts using the GMDH model. Authenticated access.
- **GMDH Education Pages:**
  - **/firstdesign:** Interactive central node with orbiting icons linking to different GMDH educational pages. (`firstdesign.html`)
  - **/introduction1:** Overview of GMDH with text and neural network diagrams. (`introduction1.html`)
  - **/architecture:** Explains GMDH architecture in six sidebar-navigated sections. (`architecture.html`)
  - **/advantages:** Highlights GMDH benefits and comparisons via tab-based navigation. (`advantages.html`)
  - **/training:** Covers training concepts like data prep, initialization, and validation. (`training.html`)
  - **/application:** Discusses GMDH applications in various sectors like weather, medicine, and finance. (`application.html`)
- **Weather Pages:** Routes include `/wind`, `/humid`, `/maxtemp`, and `/temp`, each likely rendering weather-based pages. Authenticated access.
- **Logout (/logout):** Clears the session and redirects to the landing page. Authenticated access.

### C. Technical Implementation

#### 1) Backend:

- **Framework:** Flask (Python), integrated with Flask-SQLAlchemy for ORM.
- **Database:** SQLite database (`users.db`) stores user credentials.
- **Machine Learning:**
  - Dataset: `wind_dataset.csv` including features such as RAIN, T.MAX, T.MIN, T.MIN.G, WIND.
  - Preprocessing: Includes interpolation of missing values, creation of TEMP_DIFF, feature scaling, and logarithmic transformation (`log1p`) on WIND.
  - Model: A MultilayerGMDH model with up to 100 layers and 50–500 neurons per layer, stored as `gmdh_wind_model.pkl` and `scaler.pkl`.
  - Sessions: Managed using Flask's session variables to authenticate users and maintain login state.

#### 2) Frontend:

- **Templates:** HTML5 templates with CSS and JavaScript, rendered using Jinja2.
- **Theme:** Neon aesthetic with a dark background (#121212) and cyan highlights (#00ffff). Font used: `Segoe UI`.
- **Design Features:**
  - Glowing text and box shadows
  - Hover scaling effects
  - CSS animations for floating and fading
- **Layout and Interactivity:**
  - Sidebar-based navigation in educational pages
  - Interactive orbiting circles in `firstdesign.html`
  - Tab switching via JavaScript in `advantages.html`
  - Responsive design with viewport scaling for mobile

- **Assets:** Static files include `gmdh.png` and `gmdh1.png{gmdh5.png` used for visuals, and external images in introduction pages.

### *D. Deployment*

The application runs locally using Flask's debug server on `0.0.0.0`, allowing access across devices within the same network.

### *E. Visual and Interactive Elements*

- **Aesthetics:** Dark mode interface with neon cyan glow and animated effects for a modern and engaging appearance.
- **Animations:** Includes floating, hover-scaling, and text fading to enhance visual appeal.
- **Navigation:** Uses sidebars and clickable elements for intuitive browsing.
- **Interactivity:** JavaScript-powered elements such as tab switching and orbiting buttons drive user engagement.

### *F. Conclusion*

The application successfully integrates wind speed forecasting with GMDH algorithm education. It enables users to predict wind speed based on real meteorological data using a trained GMDH model while simultaneously offering rich, interactive content that teaches the fundamentals and advantages of GMDH. The futuristic neon interface, animations, and responsive layout make the platform both educational and user-friendly.