symmetry

MDPI

Article

Symmetry-Aware Simulation and Modeling of Noise-Robust Electric Load Forecasting Using Hybrid MMPF-NARX and GA/PSO

Stylianos Pappas [1,†], Alexandros Gazis [2,†] and Nikos E. Mastorakis [1,3,*]

[1] Department of Electrical Engineering and Computer Science, Hellenic Naval Academy, Terma Chatzikyriakou, 18539 Piraeus, Greece; steliospappas@teemail.gr
[2] Department of Electrical and Computer Engineering, School of Engineering, Democritus University of Thrace, 67100 Xanthi, Greece; agazis@teemail.gr
[3] English Language Faculty of Engineering, Technical University of Sofia, 1756 Sofia, Bulgaria
* Correspondence: mastor@tu-sofia.bg
† The authors are Members of the Technical Chamber of Greece (TEE–TCG).

Abstract

Reliable electric load forecasting is an important engineering problem for power-system planning, grid stability, and mission-critical energy management. This paper presents a symmetry-aware simulation and modeling framework for medium-range electric load forecasting under noisy and uncertain operating conditions. The proposed approach combines a Multi-Model Partitioning Filter (MMPF) with Nonlinear Autoregressive Exogenous (NARX) submodels, while two adaptive optimization strategies, genetic algorithm-based resource allocation (GARA) and Particle Swarm Optimization (PSO), are used to optimize the contribution weights of the parallel predictors. The modeling process uses real commercial power-system data and evaluates the forecasting framework over April–September 2025. To simulate realistic engineering disturbances, correlated symmetric Gaussian noise is injected into the testing phase under moderate and heavy noise scenarios. The cyclic symmetry of temporal variables, such as hours and months, is preserved through unit-circle encoding, while the symmetry and asymmetry of residual error symmetric distributions are examined through scatter plot analysis. As for the context of forecasting residuals as diagnostic signals, it is important to transfer symmetry properties that can be used to evaluate the behavior of optimized predictors, along with the cyclic encoding of inputs. This means that by implementing residual-symmetry analysis, the conclusion that GARA and PSO produce concentrated, balanced, and biased errors under moderate noise and heavily correlated noise conditions can be achieved. Finally, our results show that both GARA and PSO improve the robustness of the hybrid MMPF-NARX model, but PSO consistently achieves lower MAPE values, smoother convergence, and lower computational burden. The optimal configuration is obtained with nine NARX submodels, beyond which additional model complexity offers no meaningful performance gain. Overall, the study shows that symmetry-aware modeling, adaptive optimization, and noise-based simulation can support more reliable forecasting in modern power-system engineering applications.

Keywords: adaptive weight allocation; electric load forecasting; Genetic Algorithm (GA); hybrid forecasting framework; medium-range forecasting; Multi-Model Partitioning Filter (MMPF); noise robustness; Nonlinear Autoregressive Exogenous (NARX); Particle Swarm Optimization (PSO); power-system optimization

Check for updates

Academic Editor: Vasilis K. Oikonomou

Received: 19 June 2026
Revised: 4 August 2026
Accepted: 6 August 2026
Published: 11 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

For the planning of power systems and their reliable operation, electric load forecasting is necessary. The likelihood of prediction failures, notwithstanding their actual failure, can change the content of predictions. As such, the operational/reliability factor that is embedded in predictions creates a larger buffer that requires measurement and increases costs while reducing reliability scheduling. A more accurate prediction leads to a smaller buffer and less reliability. The ARIMA approach, whether hybrid or baseline, is a common load-forecasting method [1,2]. As such, the growing use of renewable energy sources and demand-side variability might increase the nonlinear characteristics of modern grids. Moreover, recent machine-learning and ensemble load-forecasting techniques deal with the nonlinearities present in the demand variability. This is important, as it provides additional robustness against changing conditions in the grid [3,4].

Classical approaches—for example, ARIMA models—have been widely used in load-forecasting problems in the past but are no longer sufficient. Since they are mostly linear models, they show poor performance for nonlinear and asymmetrical stochastic oscillations in modern energy demand [1,5]. Many hybrid forecasting models have been proposed to address these issues. The combination of artificial neural networks with multi-variable linear regression and support vector regression via fuzzy logic has been presented in [3] and in [6]. This method achieves better results than any of its parts. An alternative paper [4] delivers a hybrid framework that utilizes intelligent optimization algorithms to fine-tune the relevant weights and select key features effective for enhancing prediction accuracy [7]. Comparative analyses demonstrate that tree-based machine-learning models, namely random forests and gradient boosting, and recurrent neural networks achieve superior performance than traditional ARIMA models under nonlinear load patterns [5,8].

To handle symmetrical and asymmetrical uncertainties in modern energy demand, ensemble-based and probabilistic forecasting methods have gained significant prominence. Recent research has shown that combining a large set of potential forecasts, each one generated with different parameter settings, improves scalability and precision, especially in medium-range forecasting [9,10]. As such, the same framework should not only be used but also expanded in implementing and integrating various predictors, such as random forests and Gradient Boosting Decision Trees. Moreover, recent studies, such as Ref. [11], have suggested that using traditional methods like quantile regression and Kernel Density Estimation for developing full load-forecasting probability symmetric distributions enable us to provide an optimal and more reliable risk evaluation than traditional classic point forecasts [12,13]. In addition, it is worth mentioning that the study [14] expands the previous notion and suggests that early economic warning signs and industrial indicators must be taken into consideration when building hybrid networks. Specifically, these feature properties are important, as they assist in maintaining high accuracy scores while ensuring that the patterns keep shifting near the critical load turning points [15].

Additionally, another important aspect to consider is the incremental adaptation of advanced state-space models, as well as the careful data preparation methods used to ensure stability of operation in smart grids. Analytically, recent studies have showcased that the use of conditional hidden semi-Markov models can be used to manage uncertain demand behaviors by monitoring and detecting random changes in electricity demand. As a result, these models can provide a real-case study's point of view in terms of their operation, as shown by [16,17], as they enable the system to split into different operating states and thus enable us to examine how long each state lasts. Specifically, although they are characterized by strong performance, they are severely associated with in-depth data cleaning, preprocessing analysis, and overfitting techniques to ensure optimal training. A typical example of this statement is when a Gaussian filter is applied to smart meter data

with an aim to provide a data cleaning method for eliminating noise from outliers or generic transmission error [1,18]. As evident from the above-mentioned examples [9,11,14,16], most contemporary power networks focus on reliability and providing fault-tolerant operation by combining optimal design principles while considering the need for continuous and flexible optimization during all cycles of operation and processing.

On the one hand, an interesting technique is Particle Swarm Optimization (PSO), which is used extensively within the industry and academia alike to tune forecasting models. Specifically, it is commonly found in ARIMA models [1]; complex Artificial Intelligence (AI), such as Neural Networks (NNs) [19–21]; and general support vector machines (SVMs) [22–24]. Specifically, most of this research showcases the various ways that PSO can assist in terms of convergence and forecasting accuracy while handling large parameter spaces efficiently. More specifically, PSO specializing in artificial NNs is of particular interest, as it can achieve superior short-term forecasting than traditional NN structures [19,20]. Additionally, PSO can be fused with SVMs to develop systems with augmented generalization models and stabler predictions [22,24]. For example, some systems with these characteristics have been implemented with gray models and wavelet-based methods that aim to improve results in cases of nonlinear behavior data samples [25]. Lastly, PSO can also be found in long-term load forecasting [26] and parameter estimation in power systems [27], as it is well-regarded and used widely due to its simplicity in usage, decreased need for a high number of hyperparameters, and rapid convergence [28].

On the other hand, even though PSO is one of the most well-known and applied hybrid methods, there has not been much talk in terms of its use with probabilistic multi-model filtering structures, such as the Multi-Model Partitioning Filter (MMPF). Specifically, the hybrid MMPF-NARX architecture principles have not been studied extensively, although there are some interesting notions, such as assigning different weights to several nonlinear submodels that run in a distributed parallel operation. An example of this can be found in genetic algorithms (GAs), where adaptive optimization processing has been applied in many works, such as in [29,30]. Analytically, GAs are defined as systems that operate by embedding crossover and mutation operators that provide global search capabilities while at the same time introducing additional structural complexity and affecting convergence system dynamics. On the contrary, one can argue that PSO uses a velocity/position update mechanism that is often responsible for smoother convergence behavior, as indicated in many works in forecasting, such as in [19,22,25].

As such, based on the above, our paper focuses on presenting two methods, one GA-based and one PSO-based, that are to be fused with the same MMPF-NARX architecture for adaptive weight optimization [31,32]. In terms of evaluating our proposed system, we employ our test on both real and commercial power-system data [33]. Specifically, we aim to showcase the methods' technical merit by monitoring their robustness as correlated Gaussian noise is introduced during our testing phases under moderate and heavy levels to simulate real-case scenarios, and issue notices on modern commercial grids.

Our findings are interesting as, on various points, they can be interpreted as potentially applicable in future mission-critical energy environments. Data quality is addressed through an adaptive filtering preprocessing stage to enhance model resilience. Analytically, it is worth noting that our comparison extends beyond the traditional testing of properties, as it includes several important technical factors, such as forecasting accuracy (MAPE); convergence behavior, e.g., the % Training MAPE per iteration; and performance consistency across seven to 10 NARX subfilter configurations. As such, we validate our technical contribution by attempting to determine whether PSO achieves a measurable benefit over GA if they are both under the same umbrella of a probabilistic hybrid forecasting framework.

Lastly, it is worth noting that the literature review shows that studies undertaken on electric load forecasting mainly focused on improving the accuracy of some specific time-series models, machine-learning predictors, probabilistic ensembles, adaptive filtering schemes, and metaheuristic optimizations. However, three common constraints are yet to be dealt with sufficiently. Initially, periodic time variables, such as hours and months, are treated as reasonably standard numerical data, without the preservation of their time invariance in an explicit sense. The integrity of many forecasting devices is studied under defined test conditions, but it is not standard practice to examine their robustness to regulated correlated noise interferences. One area where we see a plethora of studies is the effect of parallel predictors on forecast error improvements. In third, while hybrid and multi-model forecasting frameworks improve accuracy, the structured relationships between the number of parallel predictors, model complexity, and forecasting performance are not always studied. As such, the generalizations do not appear to be systematic and informative.

Due to the aforementioned points, the objective of this research study is not just limited to improving the forecasting accuracy but also in determining whether a hybrid multi-model forecasting architecture possesses temporal symmetry and overall robustness in the presence of moderate and heavily correlated noise conditions with unnecessary complications. The MMPF-NARX framework proposed in this paper fills these voids through the unit-circle encoding of cyclic variables, residual-symmetry analysis, correlated noise testing, and a systematic comparison of 7–10 NARX submodel configurations optimized through GARA and PSO. This is presented in Appendix A Table A1, which compares the reviewed literature and presents the gaps this study will fill.

2. Methods

2.1. Multi-Model Partitioning Filter (MMPF)

Another crucial adaptive estimation method to consider is the Multi-Model Partitioning Filter (MMPF), which is developed for systems with nonlinear behavior and time-changing dynamics [34]. Specifically, MMPFs operate in several parallel-running models that focus not on a unique global predictor but rather on an individual level to describe a local view of the system's behavior on which they operate. This means that the main drive behind MMPFs is to categorize the prediction task among different candidate models, like the MapReduce algorithm classifier and stages. The main difference is that their contribution is not to aggregate results but to adjust continuously based on how well they perform and try to provide an accurate forecast according to their structure. As such, they extend beyond the MapReduce paradigm, as their outputs are combined with an adaptive weighting rule that is used to determine the weight of each model, i.e., the % it will affect each step of the final algorithm of operation [35]. As such, the goal of MMPFs is the percentage of the effect of the result during a forecast instead of generic aggregation modeling.

The weighting process is probabilistic. At each iteration, the prediction error of every submodel is evaluated against the real value. Based on this residual information, a symmetric likelihood function is calculated and used to update the posterior probability associated with each model [36]. The weights are normalized to ensure that their sum remains equal to one, preserving numerical stability and interpretability.

The optimal Minimum Mean Square Error (MMSE) estimate is given by:

$$\hat{x}(k/k) = \sum_{i=1}^{M} \hat{x}(k/k;\ \theta)\ p(\theta/k)d\theta \quad (1)$$

The probabilities are calculated online in a recursive manner as follows:

$$p(\theta/k) = \frac{L(k/k;\ \theta)}{\sum_{i=1}^{M} L(k/k;\ \theta)\ p(\theta/k\ -\ 1)d\theta} \quad (2)$$

where

$$L(k/k;\ \theta) = \left|P_y(k/k-1;\theta)\right|^{\frac{1}{2}} \cdot e^{(-0.5\ \|\tilde{y}(k/k-1;\theta)\|\ \cdot P_y^{-1}(k/k-1;\theta))} \quad (3)$$

And P_y^{-1} is calculated by the EKF equations.

The basic structure of MMPFs, with an Extended Kalman Filter (EKF) implementing either ARMA or ARIMA models for forecasting purposes, is analytically presented in [37–39]. It has also been effectively utilized for adaptive electric load forecasting using empirical data [40], integrating it with support vector machines (SVMs), demonstrating substantial enhancements in predictive accuracy [41,42].

2.2. MMPF–NARX

In the proposed method, the ARMA models implemented in each EKF in Figure 1 are substituted by Nonlinear Autoregressive Exogenous (NARX) models. These models work as nonlinear predictors within the MMPF (see Figure 2).

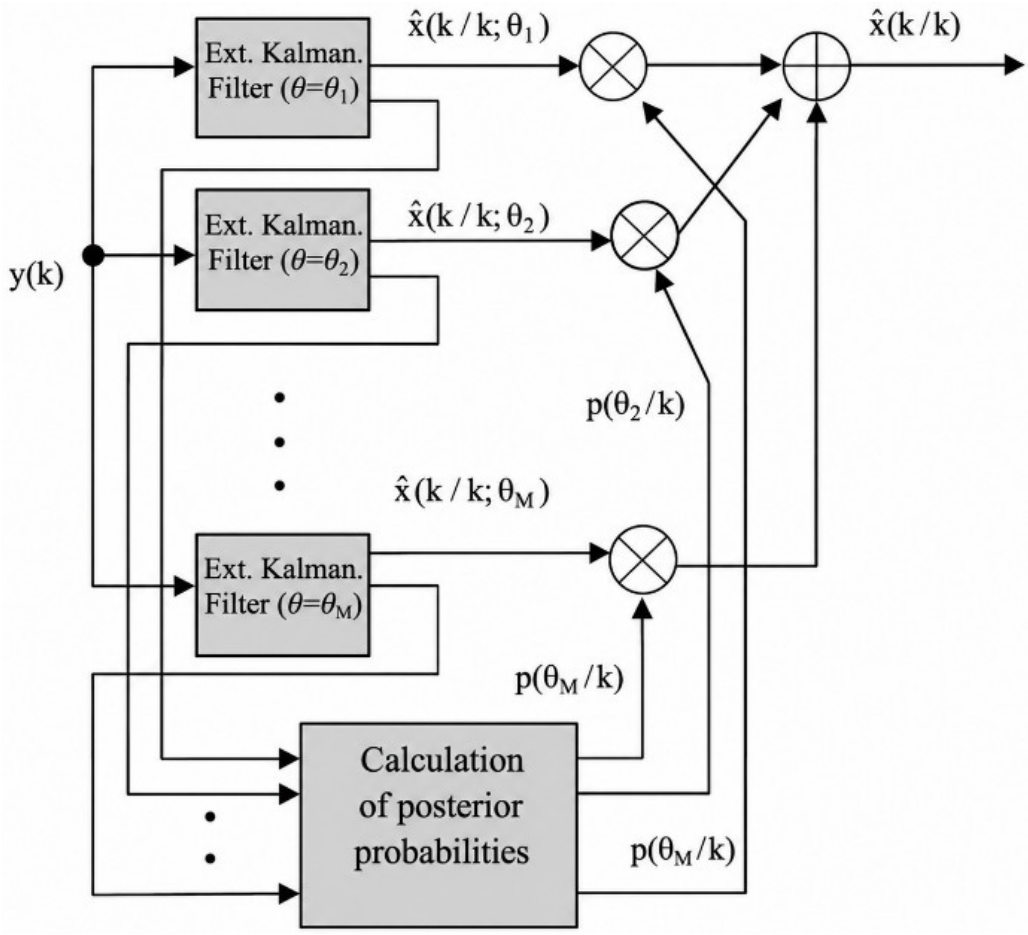


Figure 1. MMPF's basic structure.

NARX subfilters are implemented in parallel to save computational time. They aim to approximate the nonlinear relationship between the current and past electric load values, using both autoregressive and external input variables.

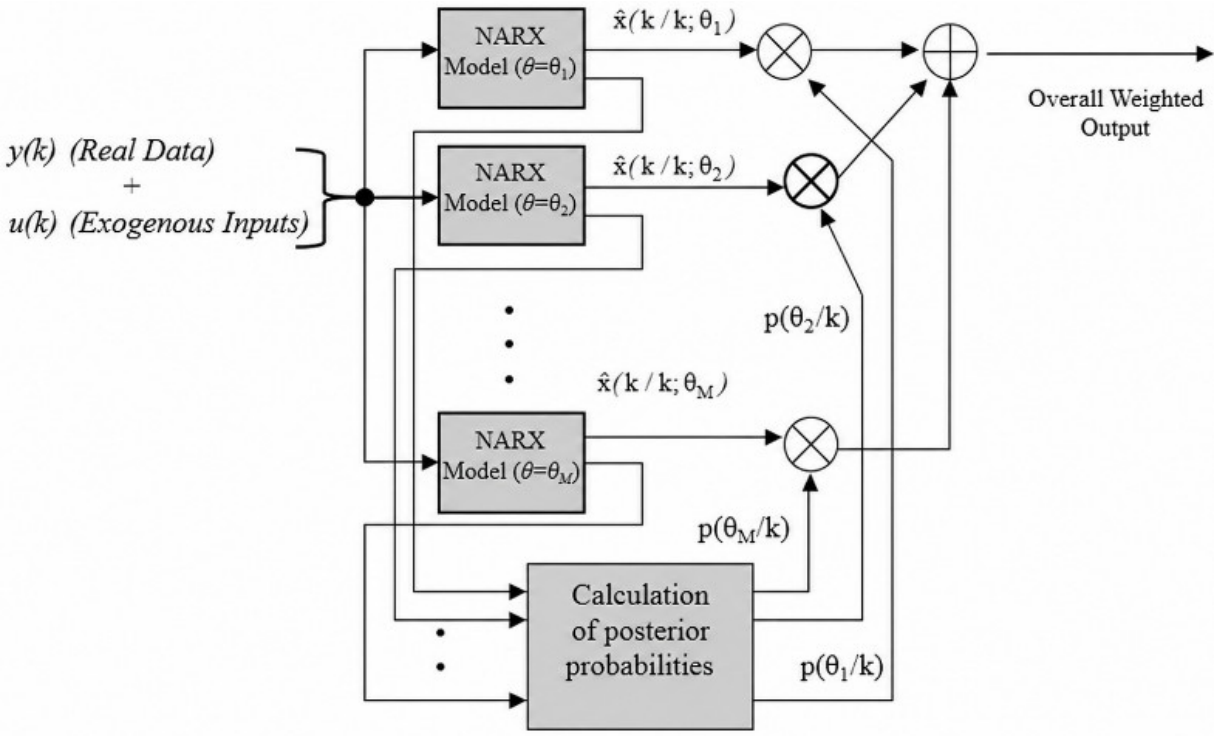


Figure 2. MMPF-NARX's architecture.

The general form of each symmetric NARX submodel is expressed as:

$$\widetilde{y}(k) = F_{\mathrm{i}}(y(k-1,(y(k-2),\ldots,y(k-p),u(k-1),\ldots,u(k-q)) \tag{4}$$

where F_{i} $(\cdot)$ represents the nonlinear vector of the ith submodel, and p and q are the autoregressive and exogenous input vector orders, respectively. The predicted electric load at time instant k, according to MMPF-NARX, is denoted as $\widetilde{y}_{\mathrm{i}}(k)$.

In this case study, the exogenous input vector $u(k)$ is defined as:

$$u(k) = \begin{bmatrix} \sin\left(\frac{2\pi D_{hour(k)}}{24}\right), \cos\left(\frac{2\pi D_{hour(k)}}{24}\right), D_{week}(k), \\ \sin\left(\frac{2\pi D_{month(k)}}{12}\right), \cos\left(\frac{2\pi D_{month(k)}}{12}\right), \overline{y}_{(k-6,k)} \end{bmatrix} \tag{5}$$

where

D_{hour} (k) = $\left[\sin\left(\frac{2\pi D_{hour(k)}}{24}\right), \cos\left(\frac{2\pi D_{hour}(k)}{24}\right)\right]$ is the cyclic encoding of the hour of the day (1–24).

D_{month} (k) = $\left[\sin\left(\frac{2\pi D_{month(k)}}{12}\right), \cos\left(\frac{2\pi D_{month(k)}}{12}\right)\right]$ is the cyclic encoding of the month of the year (1–12).

Both are expressed in a unit circle.

This encoding prevents discontinuities (i.e., from 23:00 to 0:00, or from the 12th month to the 1st). It effectively models the diurnal pattern of energy consumption.

D_{week} (k) = the day of the week (1–7).

$\overline{y}_{(k-6,k)}$ = the moving average of the load, over the past six hours, and is defined as:

$$\frac{1}{6}\sum_{j=k-6}^{k-1} y(j)$$

Lastly, each NARX submodel minimizes its own mean-squared error:

$$J_{i=}\frac{1}{M}\sum_{k=1}^{M}\left(y(k) - \widetilde{y}_i(k)\right)^2 \tag{6}$$

2.3. GARA

To improve the adaptive weight selection of the MMPF-NARX structure, a genetic algorithm-based resource allocation (GARA) mechanism is added. The genetic algorithm works with different candidate weight vectors. It optimizes them by minimizing a fitness function based on forecasting error [43]. Through selection, crossover, and mutation procedures, the population gradually moves toward an optimal combination of NARX submodels, refining the probabilistic weighting provided by the MMPF [44].

The input to the genetic algorithm is the overall weighted output, as estimated by MMPF-NARX.

Each chromosome represents a candidate weight vector:

$$W = [w_1, w_2, \ldots w_M] \tag{7}$$

with $\sum_{i=1}^{M} w_i = 1$, and $w_i \in [0,1,]$,

where M denotes the number of NARX submodels (7–10, depending on the configuration).

The first chromosome is initialized from the final vector of posterior probabilities as it is estimated by the MMPF:

$$w^1 = w^{MMPF\text{-}NARX} = [p(\theta_1/k) \ldots . p(\theta_M/k)] \tag{8}$$

with $\sum_{i=1}^{M} w_i = 1$, and $w_i \in [0, 1]$.

The fitness function measuring the forecasting accuracy is given by:

$$f(\boldsymbol{W}) = \frac{1}{N}\sum_{k=1}^{N}\left|y(k) - \sum_{i=1}^{M} w_i \widetilde{y}_i(k)\right|^2 \tag{9}$$

The genetic algorithm evolves the population via selection, crossover, and mutation until convergence. The mutation rate is defined as:

$$\mu_t = \mu_0\left(1 - \frac{t}{t_{max}}\right) + \varepsilon \tag{10}$$

where

- μ_t = the mutation rate at generation t;
- μ_0 = the initial mutation rate;
- t_{max} = the maximum number of generations;
- ε = the residual mutation term, ensuring minimal diversity near convergence.

In this study, μ_0 is set to 0.1. This corresponds to a moderate mutation rate of 10%, which helps to avoid early convergence and keeps a good level of exploration. The weights assigned to each NARX submodel are optimized adaptively, so t_{max} was chosen as 40. Finally, ε is set to 0.003 to introduce a small stochastic disturbance to maintain diversity near convergence and avoid stagnation when fitness stops improving, without reducing stability.

The optimal weight vector is given by:

$$\boldsymbol{W}^* = argmin\ f(\boldsymbol{W}) \tag{11}$$

And the final optimized forecast is expressed by:

$$\hat{y}\ (k/k) = \sum_{i=1}^{M} w_i^* \widetilde{y}_i(k) \tag{12}$$

where w_i^* represents the optimal weight assigned to the ith NARX submodel.

2.4. PSO

To examine whether swarm-based optimization offers measurable advantages over evolutionary resource allocation, the GARA mechanism is replaced by a Particle Swarm Optimization layer while preserving the same hybrid MMPF–NARX structure.

The input of the Particle Swarm Optimization (PSO) algorithm is the weighted prediction structure produced by the MMPF–NARX framework. Each particle represents a candidate weight vector:

$$W = [w_1, w_2, \ldots w_M] \tag{13}$$

with $\sum_{i=1}^{M}(w_i) = 1$, and $w_i \in [0, 1]$,

where M denotes the number of NARX submodels (7–10 depending on the configuration).

In this work, the swarm consists of $N_p = 25$ particles, which is sufficient given the relatively low dimensionality of the weight vector. To preserve consistency with the probabilistic structure of the MMPF–NARX framework, a hybrid initialization strategy is adopted.

The first particle is initialized using the posterior probability vector estimated by the MMPF:

$$x_1^0 = w^{MMPF\text{-}NARX} = [p(\theta_1/k \ldots p)\theta_M/k \tag{14}$$

The remaining x_j^0 are randomly generated as $w_i \sim U(0, 1)$ within the admissible region and subsequently normalized to satisfy the unity-sum constraint $\sum_{i=1}^{M} w_i$, ensuring that each particle represents a valid convex combination.

Initial velocities are randomly assigned within a small symmetric interval $v_j^0 \sim U(-v_{max}, v_{max})$, where v_{max} was set to 0.1 to prevent excessive oscillatory behavior and ensure stable exploration within the bounded weight space.

The forecasting accuracy is evaluated using the same objective function defined for the GA approach with Equation (9).

At iteration t, the velocity of the j-th particle is updated as:

$$v_j^{t+1} = \omega v_j^t + c_1 r_1 \left(\mathbf{p}_j^t - \mathbf{x}_j^t\right) + c_2 r_2 \left(\mathbf{g}_j^t - \mathbf{x}_j^t\right) \tag{15}$$

where

- ω = the inertia weight;
- c_1 and c_2 = the acceleration coefficients;
- r_1 and $r_2 \sim U(0, 1)$;
- $\mathbf{p}_j^t$ = the personal best position of particle j;
- $\mathbf{g}_j^t$ = the global best position of the swarm.

For this work, $\omega = 0.7$ and $c_1 = c_2 = 1.5$, ensuring a balanced exploration and exploitation trade-off.

As such, the position of each particle is updated according to

$$\mathbf{x}_j^{t+1} = \mathbf{x}_j^t + v_j^{t+1} \tag{16}$$

After each update, the negative weights are clipped to zero, the weights greater than one are reduced to one, and the vector is renormalized as:

$$w_i^{t+1} = \frac{w_i^{t+1}}{\sum_{i=1}^{M} w_i^{t+1}} \tag{17}$$

The PSO algorithm terminates when either the maximum number of iterations ($t_{max} = 40$) is reached or when the improvement in the global best fitness $| f_{best}^{t+1} - f_{best}^{t} |$ becomes less than 10^{-6}. This tolerance ensures convergence stability without enforcing excessive computational effort.

The final optimal weight vector and the final optimal forecast are given by Equations (11) and (12), as given for the GARA case.

2.5. Data Preparation

The dataset covers the period from January 2020 to December 2025. It is arranged in chronological order to keep the time structure of the load data. The data are split sequentially. Approximately 60% is used to train the MMPF–NARX submodels, 20% is used for validation, and the remaining 20% is reserved for testing and forecast evaluation.

To check how the proposed model works, anomalous measurements were inputted, in which a disturbance was added only to the test subset. Specifically, we focused on providing a distinct process in terms of training and validation, i.e., to ensure that, in all stages, the model deployed does not learn from noisy, incomplete, or generally corrupted data. As such, to achieve this, correlated Gaussian noise was chosen to be added to our testing phases in an effort to simulate realistic measurement errors and sensor uncertainty behavior. The full data flow of the proposed MMPF-NARX framework, including the transition from weighted model outputs to the GARA or PSO stage, is summarized in Figure 3.

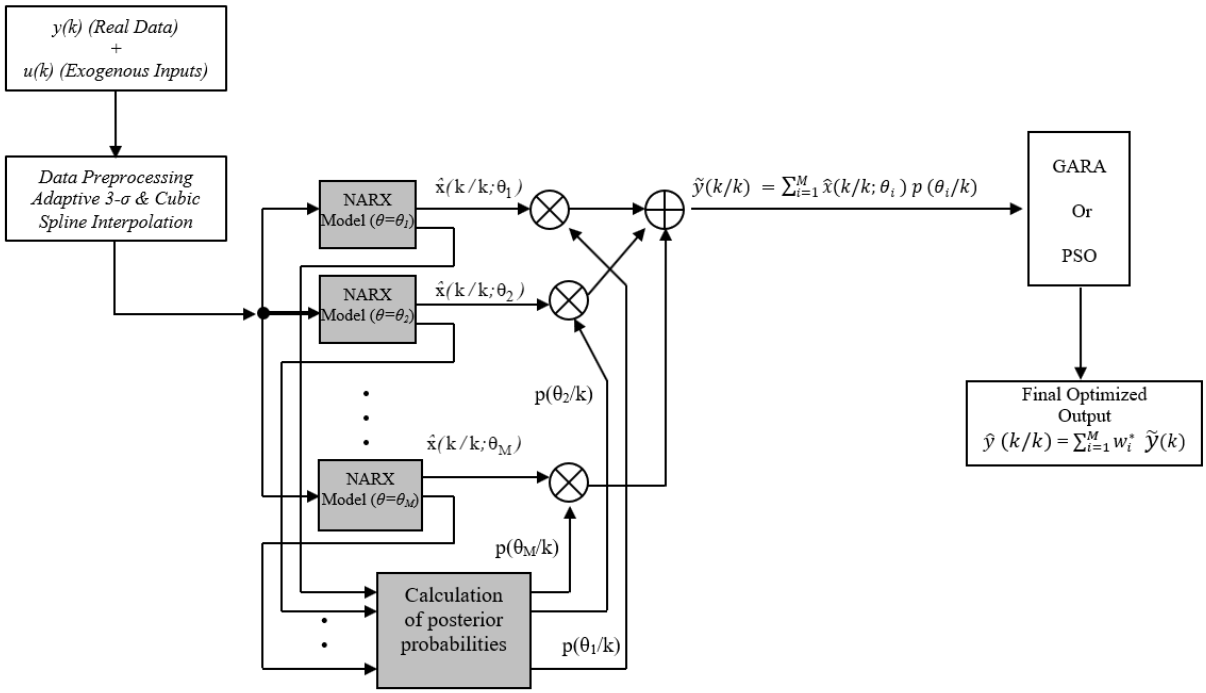


Figure 3. MMPF-NARX's data flow. Final weighted outputs become inputs for either GARA or PSO for further processing and improvement.

Before feeding the raw load data into the MMPF-NARX framework, a data cleaning process takes place. It is known that power grids' real data are often corrupted or have missing values due to sensor failures, communication lag, or maintenance. To ensure the accuracy of the forecasting models, we implement a two-step preprocessing strategy.

The traditional 3σ (three-sigma) rule assumes a static mean and standard deviation. However, electric load is non-stationary and follows daily and seasonal patterns. There-

fore, we adopt an improved 3σ criterion based on an Exponentially Weighted Moving Average (EWMA).

Instead of a fixed global mean, we calculate a local moving average that adapts to the load's trend. If a measurement $z(k)$ satisfies the following condition, it is flagged as an outlier:

$$|z(k) - \mu_{\text{local}}(k)| > 3\sigma_{\text{local}}(k) \tag{18}$$

where μ_{local} and σ_{local} represent the adaptive means and standard deviation at time k. As such, errors should not take into account natural peak loads (midday peaks, for example) while removing technical faults or spikes [45].

When outliers are removed, or missing intervals are found, there will be gaps in the time series that need to be filled. As such, we can use Cubic Spline Interpolation for this purpose. This means that the cubic spline connects the data points using third-order polynomials. As such, this approach guarantees the continuity of both the first and second derivatives of the load curve. Also, the continuity of symmetric signals is very important for the NARX subfilters, as it enables the internal Kalman filters to keep on smoothly interpolating the subfilter state, thus avoiding the prediction error exploding due to discontinuities in the artificial data [46,47].

The forecasting testing period was selected as follows: January–February (Period 1), April–May (Period 2), and August–September (Period 3). These snippets exhibit different seasonal patterns. They also allow for a transparent assessment of medium-term operational conditions.

To test the model's robustness, two levels of correlated noise were selected:

- *Moderately correlated noise:* Correlation coefficient $\rho = 0.95$ and noise ratio = 0.25.
- *Highly correlated noise:* Correlation coefficient $\rho = 0.98$ and noise ratio = 0.50.

Each of these scenarios simulates the distortion of the data at various levels. They allow us to visualize the performance change when more challenging conditions are added.

The proposed framework's arrangement was also tested by altering the number of parallel NARX submodels. Configurations consisting of 7, 8, 9, and 10 submodels were implemented and evaluated under each noise level. This comparison shows how the number of submodels, i.e., the proposed method's complexity, affects forecasting accuracy, robustness characteristics, and error behavior when measurement disturbances are present.

To assess the accuracy of the predictions, the Mean Absolute Percentage Error (MAPE) is calculated as follows:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{A_i - P_i}{A_i} \right| \tag{19}$$

where A_i is the actual value, P_i is the predicted value, and n is the total number of samples.

The percentage Training MAPE per iteration is used to examine the convergence behavior of the optimization algorithms.

3. Results

As has been mentioned, the real data was injected with noise. Figures 4 and 5 depict the effect of moderate and heavy noise on the data series. Similarly, Figures 6 and 7 present the forecasting performance of the PSO-based model under moderate and heavy noise, while Figures 8 and 9 present the corresponding forecasting performance of the GARA-based model under the same noise conditions. Moreover, Figures 10–12 show the PSO error scatter plots for Period 1, Period 2, and Period 3 under moderate noise, while Figures 13–15 show the same PSO error behavior under heavy noise. Furthermore, Figures 16–18 present the GARA error scatter plots for the three periods under moderate noise, while Figures 19–21 present the corresponding GARA error scatter plots under heavy

noise. Lastly, Figures 22–24 compare the fitness training error per iteration of both methods under moderate noise, while Figures 25–27 compare their fitness training error per iteration under heavy noise.

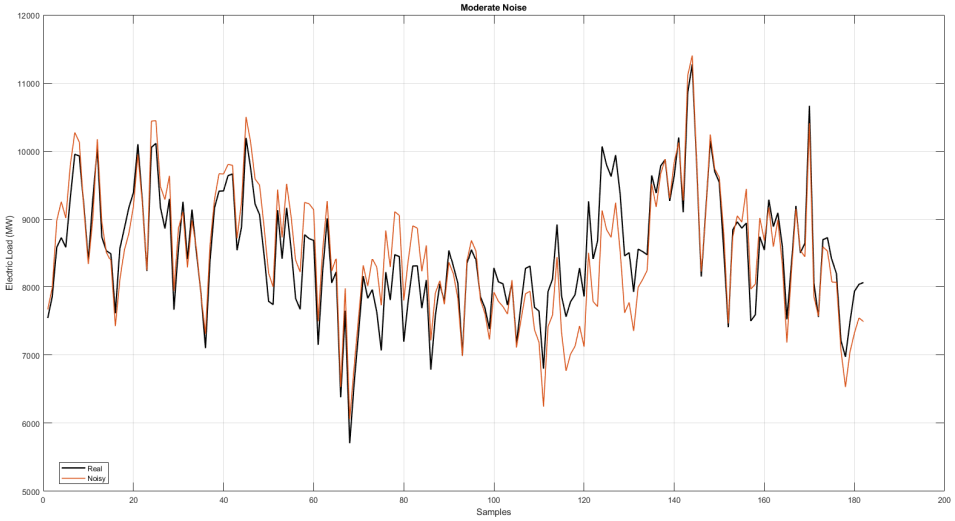


Figure 4. Moderate noise scenario; correlation coefficient ρ = 0.95 and noise ratio = 0.25.

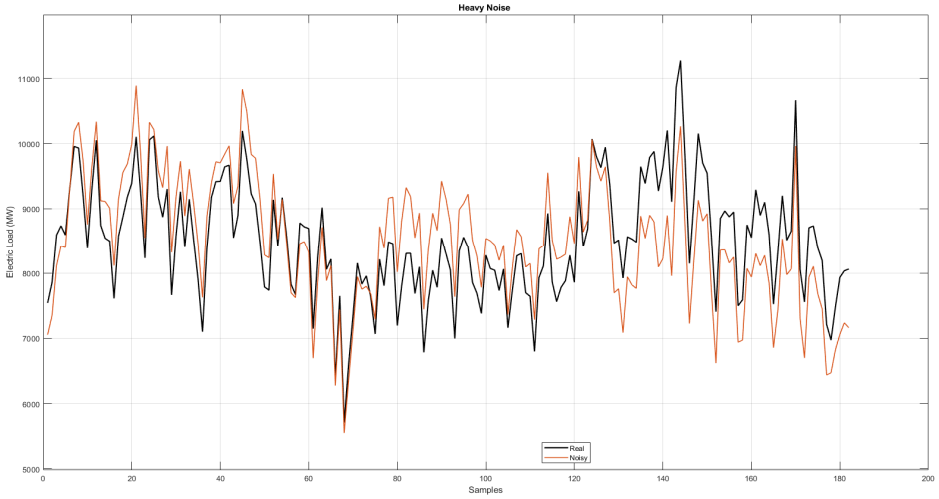


Figure 5. Heavy noise scenario; correlation coefficient ρ = 0.98 and noise ratio = 0.5.

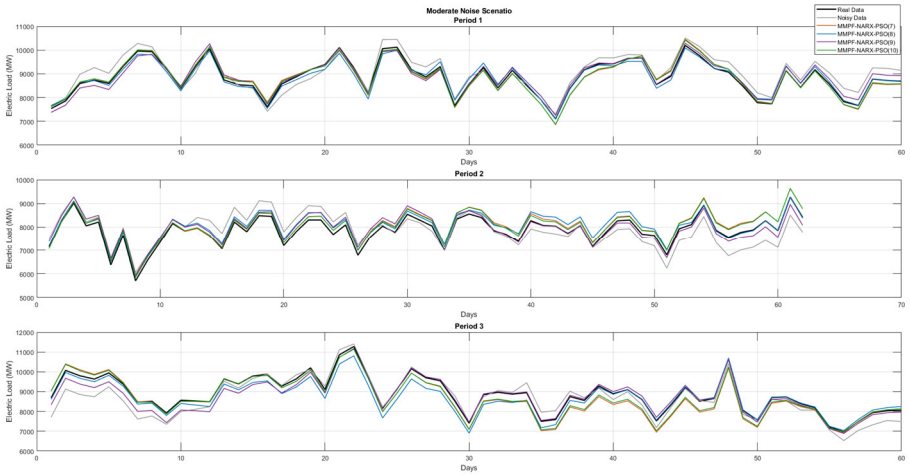


Figure 6. Forecasting performance using PSO under the moderate noise scenario.

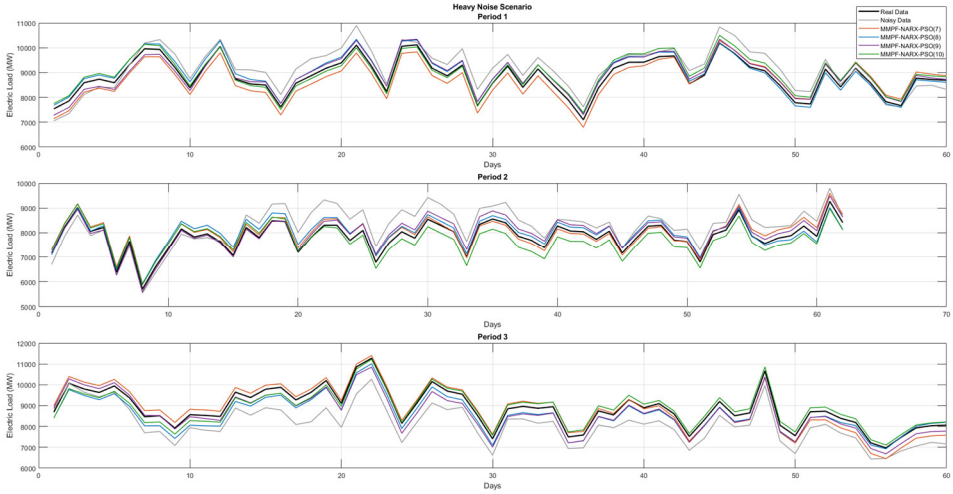


Figure 7. Forecasting performance using PSO under the heavy noise scenario.

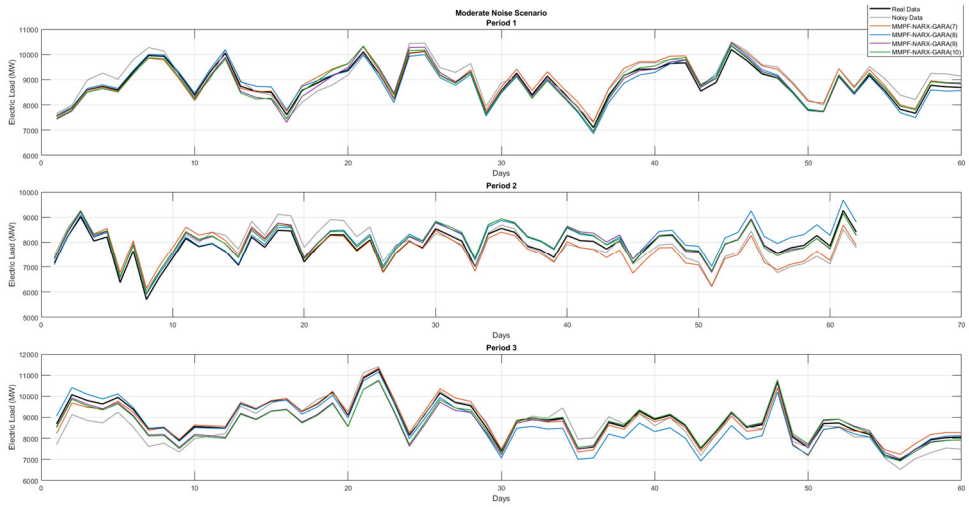


Figure 8. Forecasting performance using GARA under the moderate noise scenario.

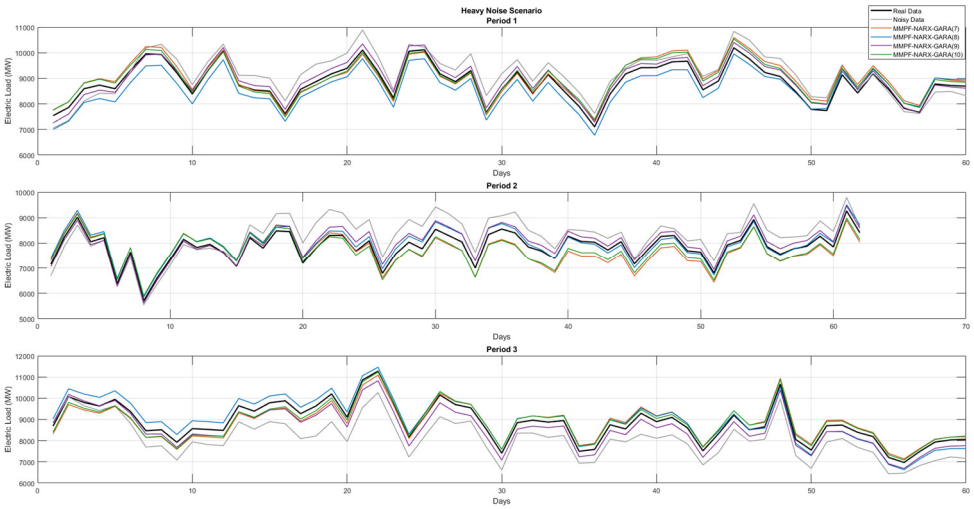


Figure 9. Forecasting performance using GARA under the heavy noise scenario.

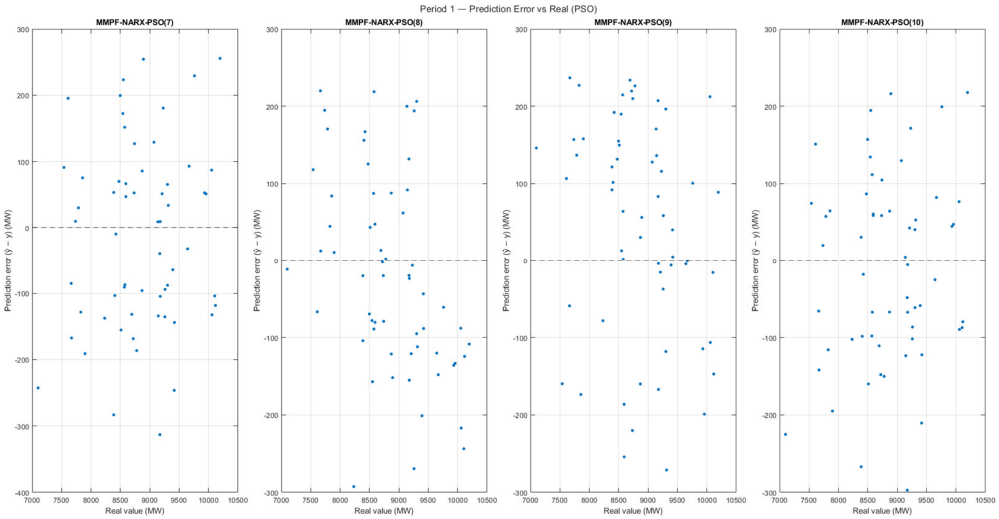


Figure 10. Moderate noise scenario. PSO error scatter plot, Period 1.

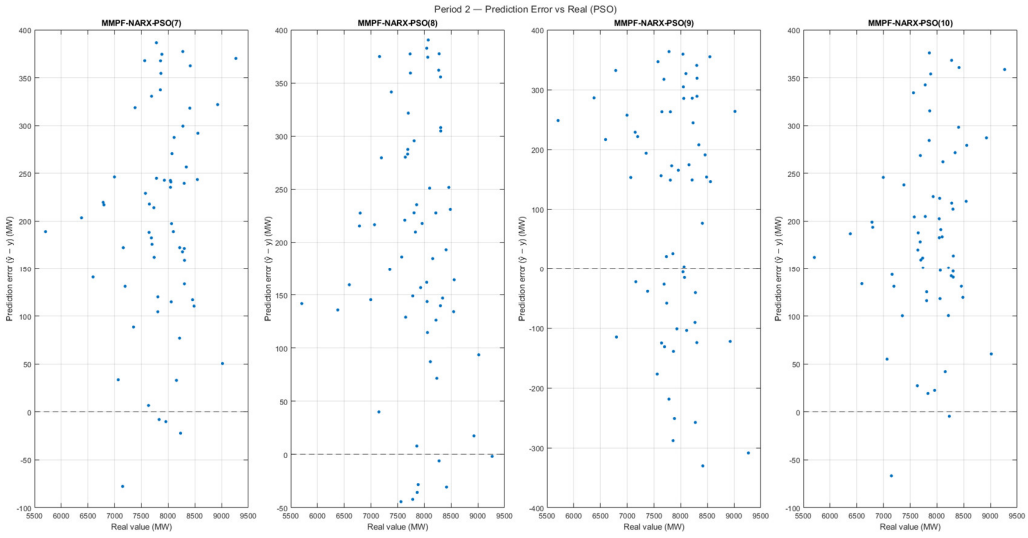


Figure 11. Moderate noise scenario. PSO error scatter plot, Period 2.

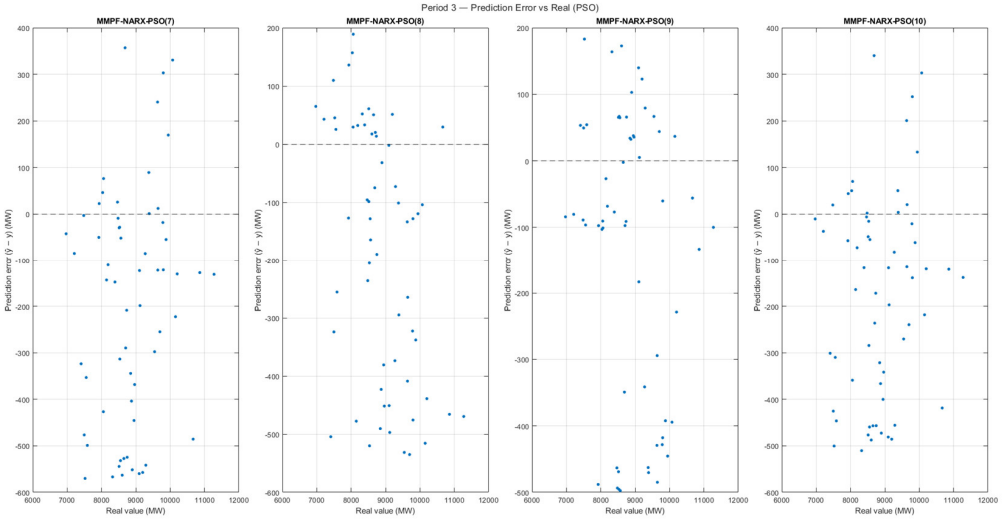


Figure 12. Moderate noise scenario. PSO error scatter plot, Period 3.

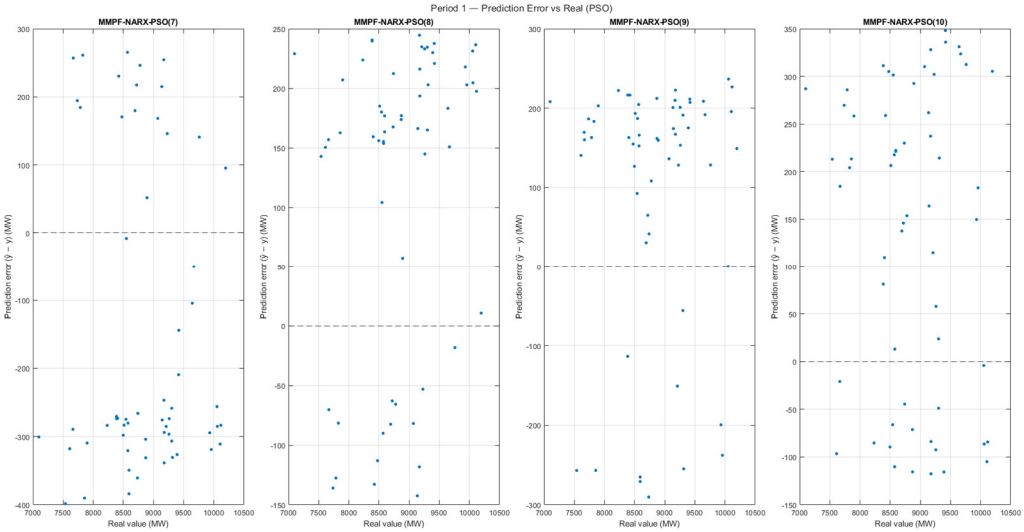


Figure 13. Heavy noise scenario. PSO error scatter plot, Period 1.

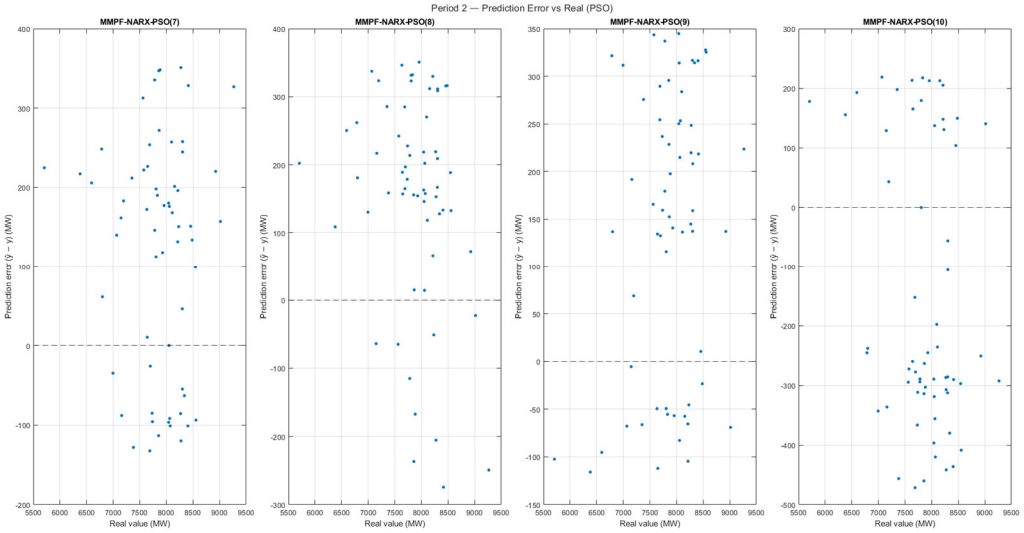


Figure 14. Heavy noise scenario. PSO error scatter plot, Period 2.

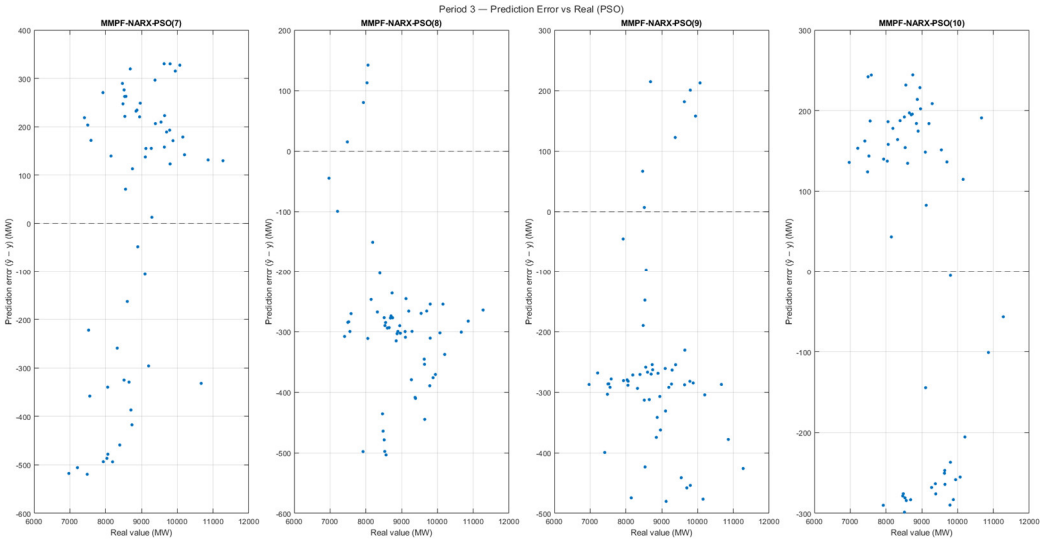


Figure 15. Heavy noise scenario. PSO error scatter plot, Period 3.

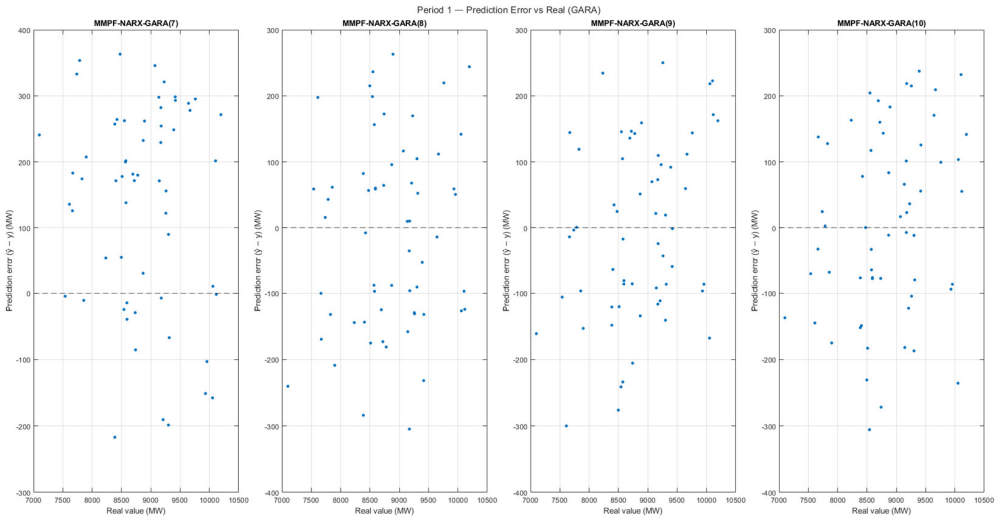


Figure 16. Moderate noise scenario. GARA error scatter plot, Period 1.

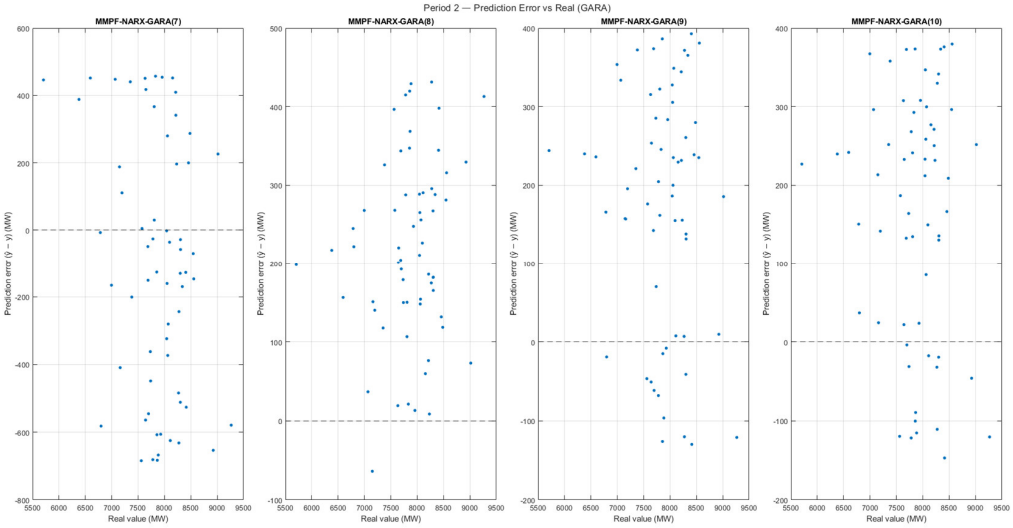


Figure 17. Moderate noise scenario. GARA error scatter plot, Period 2.

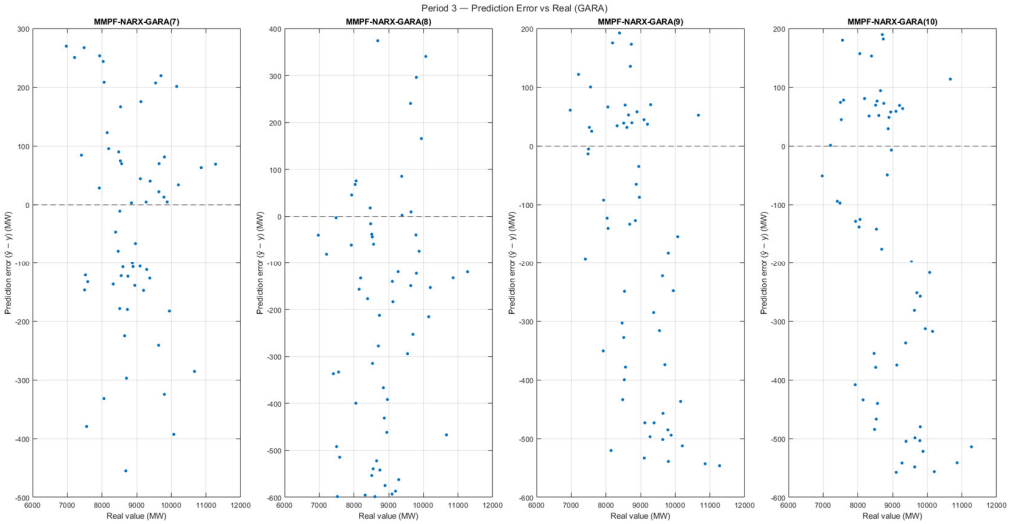


Figure 18. Moderate noise scenario. GARA error scatter plot, Period 3.

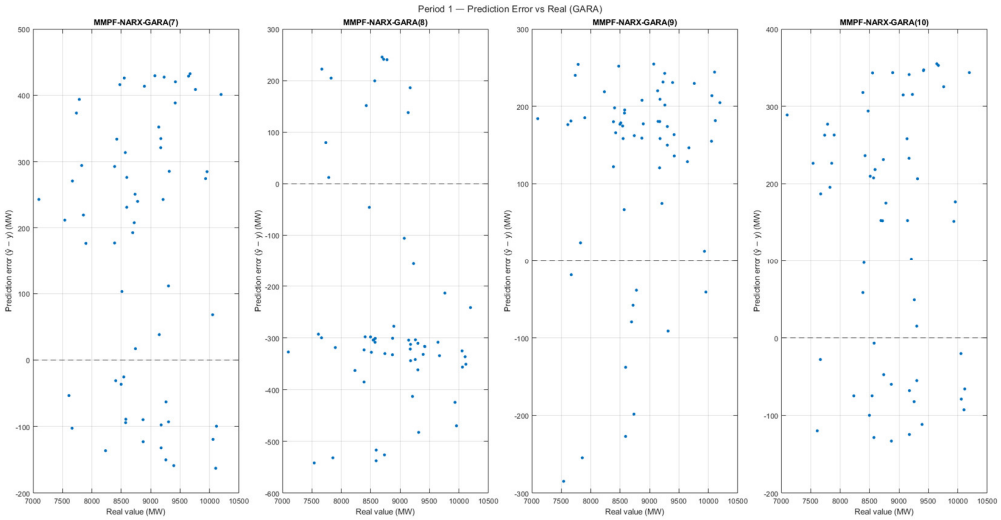


Figure 19. Heavy noise scenario. GARA error scatter plot, Period 1.

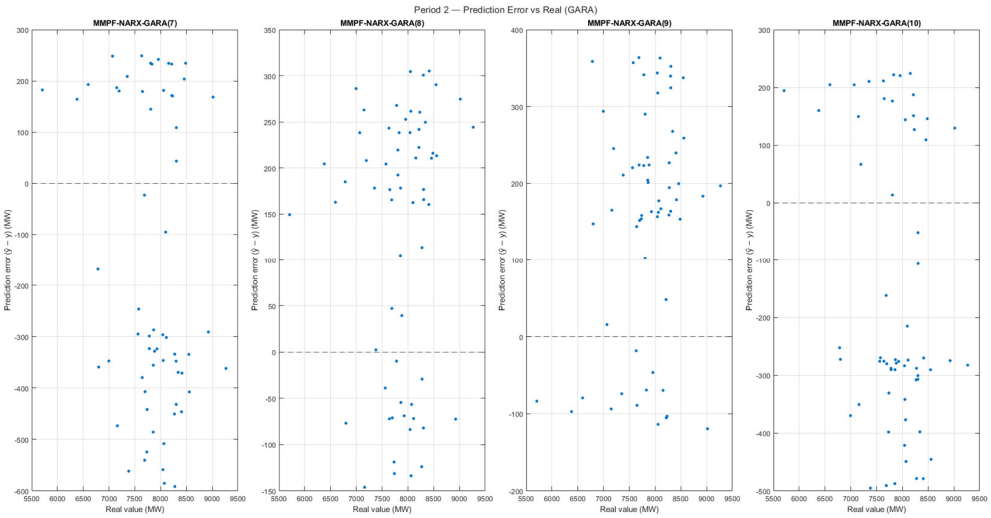


Figure 20. Heavy noise scenario. GARA error scatter plot, Period 2.

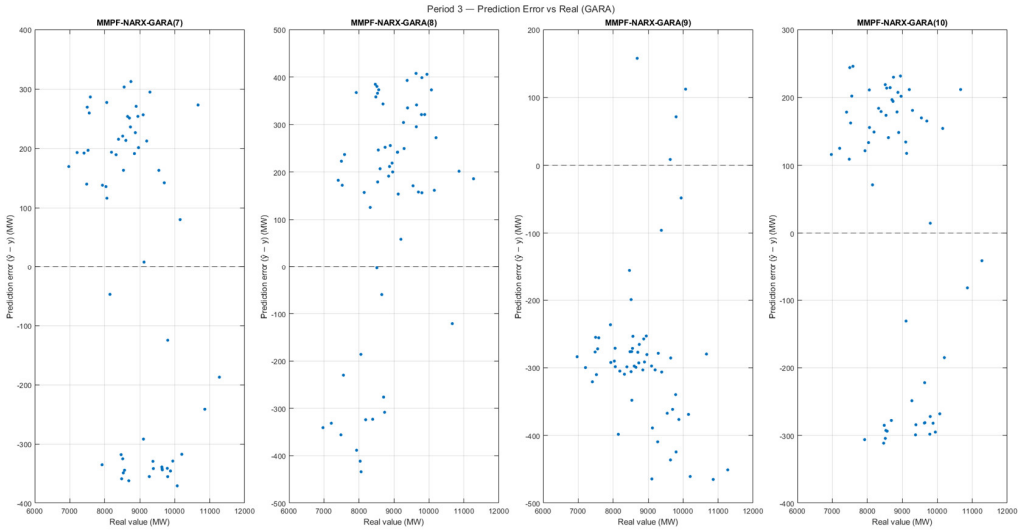


Figure 21. Heavy noise scenario. GARA error scatter plot, Period 3.

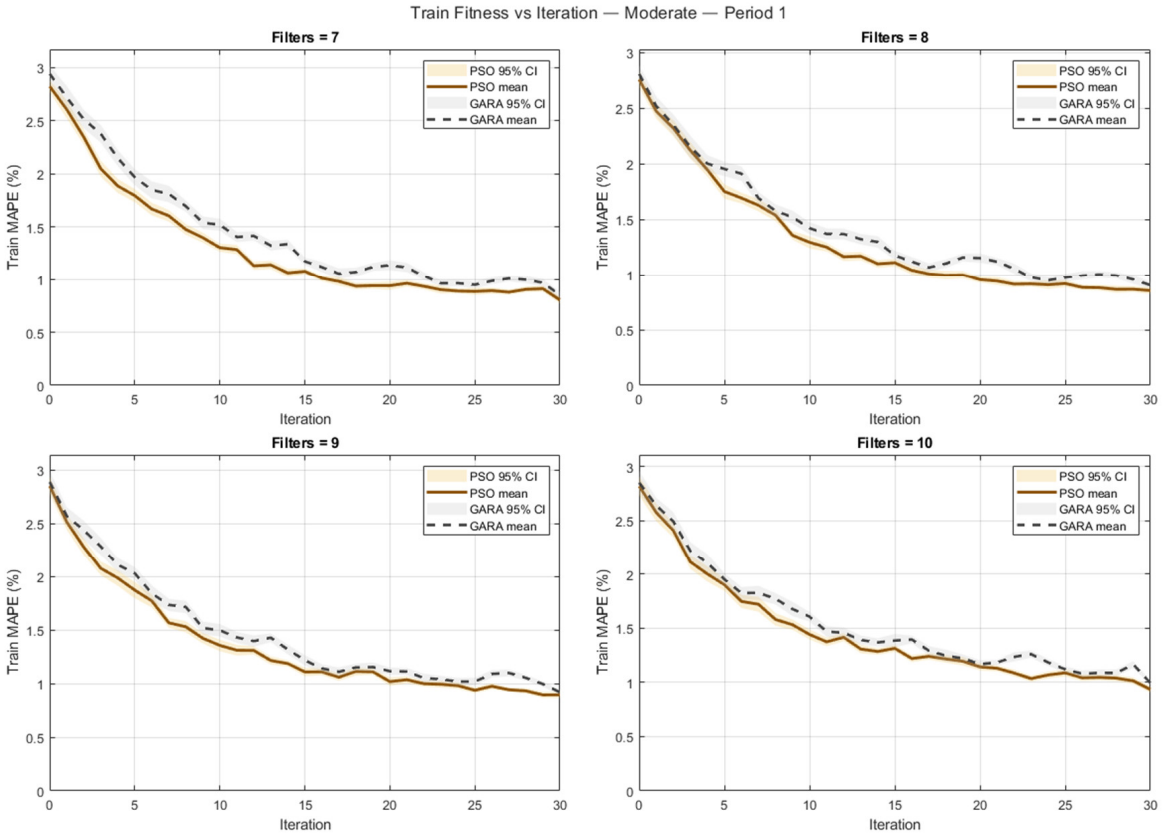


Figure 22. Comparison of fitness training error per iteration for both methods. Moderate noise scenario, Period 1.

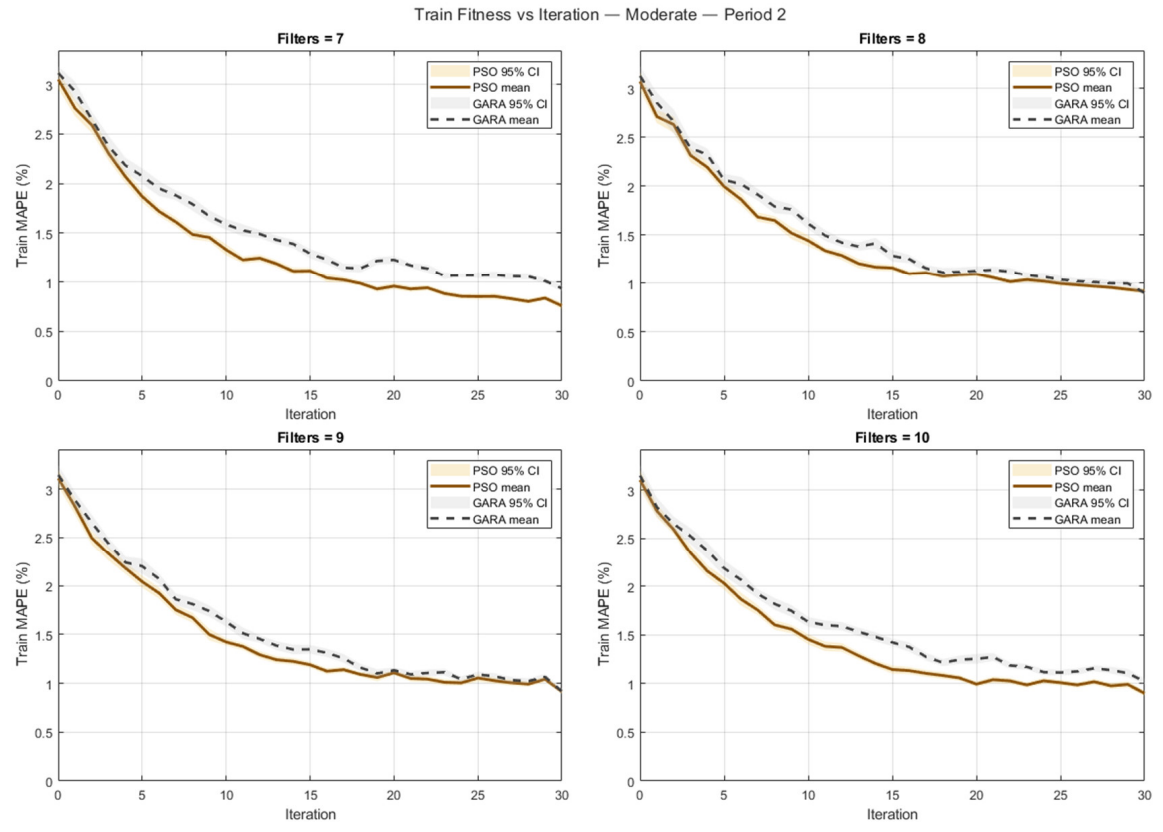


Figure 23. Comparison of fitness training error per iteration for both methods. Moderate noise scenario, Period 2.

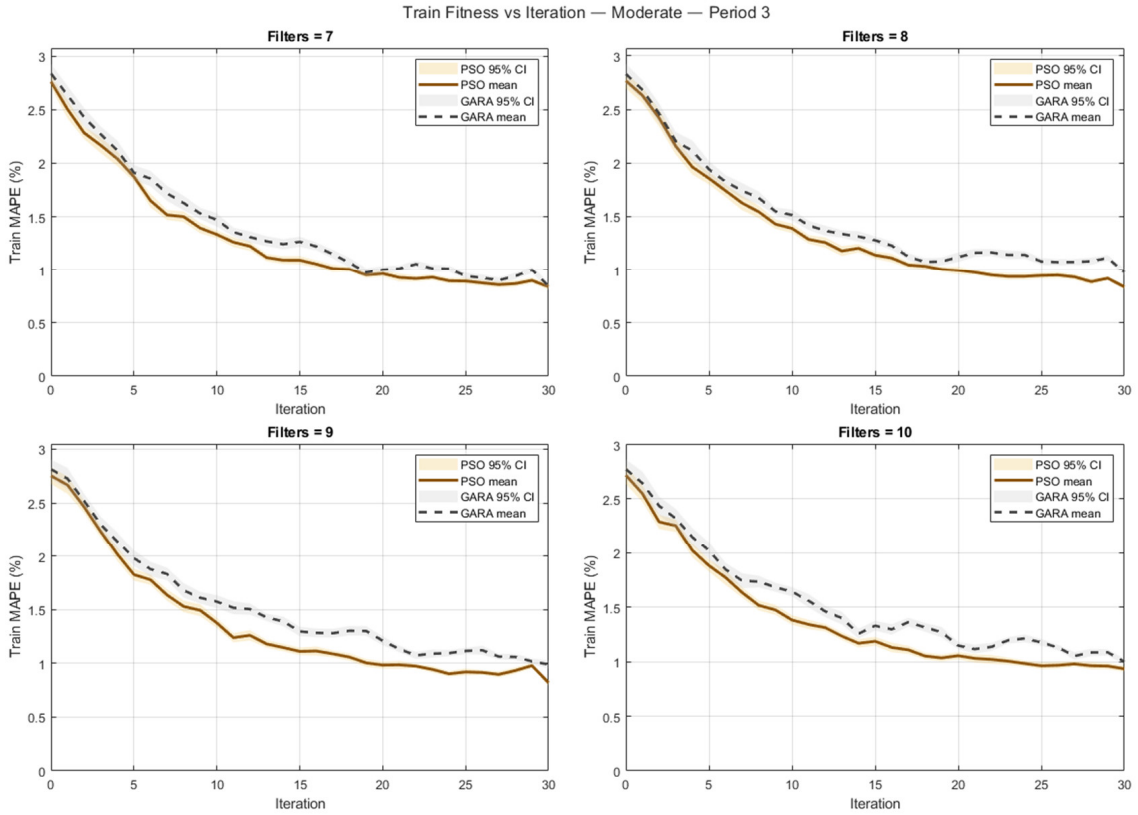


Figure 24. Comparison of fitness training error per iteration for both methods. Moderate noise scenario, Period 3.

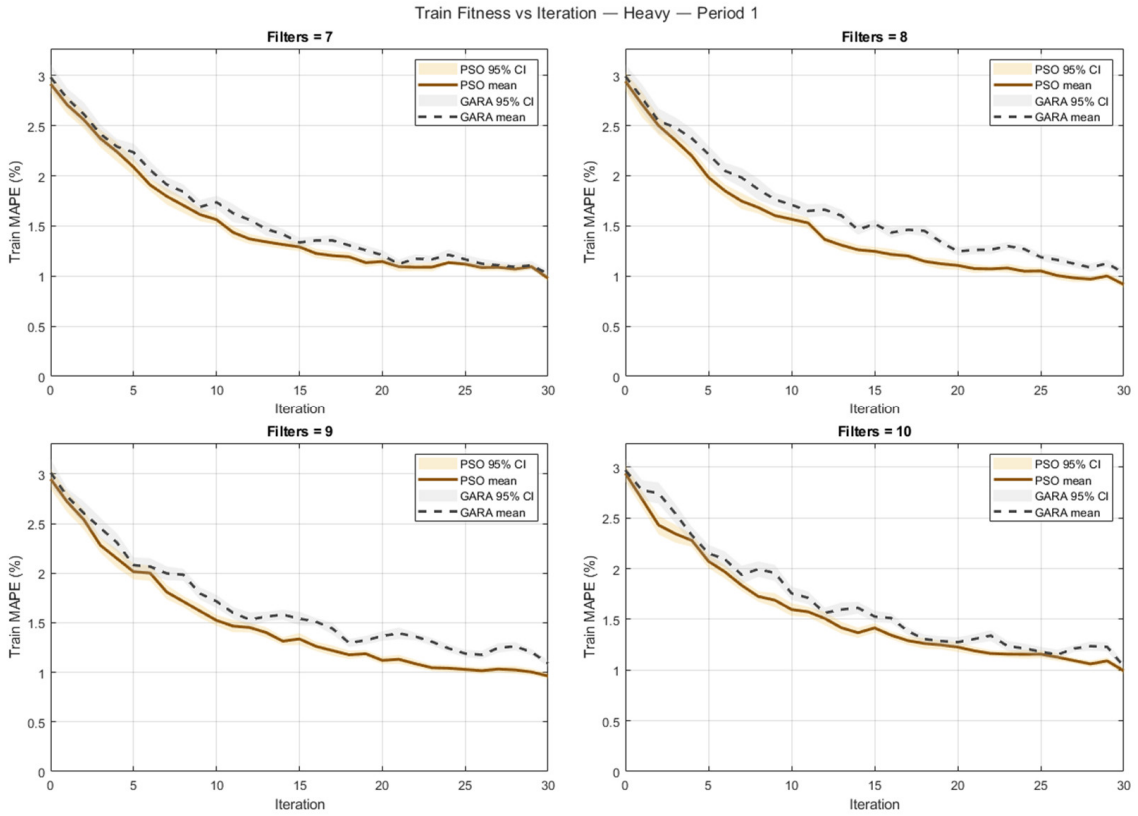


Figure 25. Comparison of fitness training error per iteration for both methods. Heavy noise scenario, Period 1.

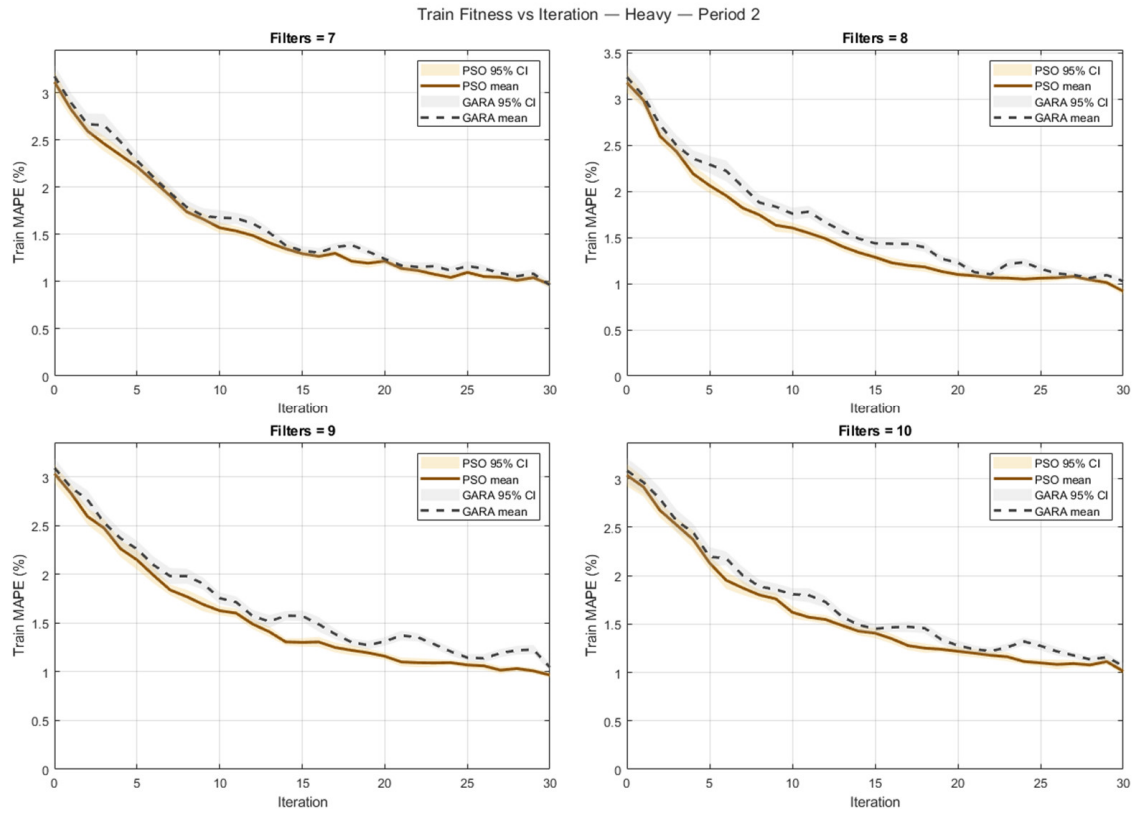


Figure 26. Comparison of fitness training error per iteration for both methods. Heavy noise scenario, Period 2.

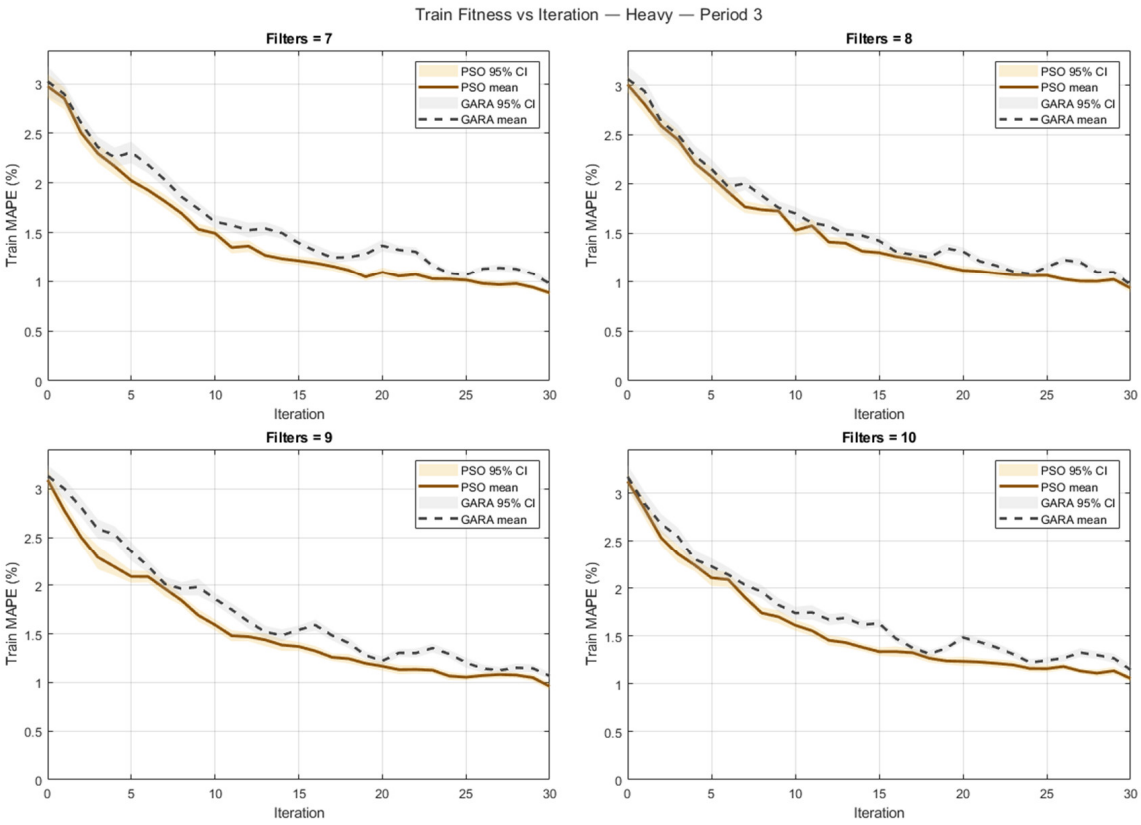


Figure 27. Comparison of fitness training error per iteration for both methods. Heavy noise scenario, Period 3.

The two methods were tested for day-ahead prediction in a timeframe of two months. Three different forecasting periods were selected. Period 1 corresponds to June–July 2025, Period 2 to April May 2025, and Period 3 to August–September 2025.

4. Discussion

4.1. Comparative Evaluation of Optimization Strategies

The experimental results presented in Table 1 show that both GARA and PSO give stable and reliable performance within the hybrid MMPF-NARX model. Analytically, our results are promising, since during all our test cases, our two methods employed achieved low MAPE values, thus showcasing their suitability for medium-term load forecasting in complex and commercial grid conditions.

Table 1. An overview of our MAPEs for all the employed methods (including different noise scenarios).

Noise Scenario	# of NARX Models (N)	MMPF-NARX		Behavior
		GARA	PSO	
Medium	7	2.70%	2.30%	Under-Capacity
	8	2.42%	2.08%	Improving
	9	2.15%	2.02%	Optimal
	10	2.17%	2.06%	Plateau
Heavy	7	3.1%	2.88%	Under-Capacity
	8	2.88%	2.58%	Improving
	9	2.55%	2.49%	Optimal
	10	2.61%	2.53%	Plateau

However, a closer examination of the metrics reveals that the proposed PSO method maintains a consistently better performance in all configurations. Specifically:

- The overall forecast accuracy regarding the noise level and the number of submodels implemented rose from 2.4% to 14.8%. This is important in terms of the % MAPE for PSO over GARA.
- In the "*medium noise*" optimal configuration (i.e., in Table 1 for N = 9), PSO reached a value of 6.0%. This is important in terms of relative improvement, i.e., PSO's 2.02% is comparable to 2.15% for GARA, as indicated in Table 1.
- Similarly, in the "*heavy noise*" optimal configuration (i.e., in Table 1 for N = 9), PSO achieved a reduced 2.4% forecasting error, indicating an accuracy improvement.

In conclusion, based on the above, our findings showcase that using PSO algorithms can provide a more efficient fine-tuning submodule weight system. This is particularly important in cases where even higher noise level conditions occur; GARA remains a robust resource allocation method when under noise.

4.2. Non-Monotonic Relationship Between Filter Complexity and Accuracy

An important finding of our study is the connection between the number of parallel NARX subfilters (N) and the forecasting accuracy results, which are *not monotonic*. Analytically, as presented in Table 1, the increased complexity is not connected to a sudden shift or a continuous reduction in error. On the contrary, Table 2 suggests that hybrid models can employ three different types of behavior.

As evident from Table 2, the NARX filter value exceeds an optimal value N (in this case, $N = 9$) as the method tries to either over-parameterize or start modeling the noise outliers as a symmetric signal. As a result, this behavior leads to reduced forecasting performance due to noise values passing through our initial system to the forecasting analysis.

Table 2. An overview of NARX models (N) on forecasting performance, i.e., a description of the behavior based on different cases.

Case	# of NARX Models (N)	Behavior Description
1. **Complexity Deficit**	7, 8	Higher error rates are observed. The model cannot fully capture nonlinear oscillations and measurement noise. The number of subfilters is not enough.
2. **Optimal Complexity**	9	Best performance for both optimization methods. The diversity of subfilters is well-balanced. This gives a more accurate representation of load dynamics.
3. **Performance Plateau and Degradation**	10	Error increases slightly (e.g., MAPE rises from 2.49% to 2.53% under heavy noise with PSO). Extra filters add unnecessary parameters. This may cause overfitting and reduce generalization ability.

4.3. Robustness and Practical Considerations

In terms of robustness and practical considerations, the MMPF-NARX framework is an ideal solution derived from the results of our testing phases for various forecasting periods and noise levels. Specifically, a performance gap between PSO and GARA is indicated under heavy noise conditions and can be translated as a need for a swarm-based, optimization-resilient solution. This could be beneficial for systems employing critical operations in energy environments characterized by a high percentage of data uncertainty. Lastly, in terms of our PSO training, the smoother convergence curves, in association with better accuracy values, indicate that our proposed system can be used as a rapid prototype or an efficient alternative in real-time application scenarios and test cases requiring high-precision forecasts due to sensory device-related uncertainties.

4.4. Integration into Real-World Grid Operations: Computational Feasibility and Scalability

The MMPF-NARX framework proposed aims to perform two complementary functions in a grid operation setting: network state forecasting and decision support for control actions managing main grid disturbances. In a practical situation, this means that it shall obtain historical and live data from SCADA systems, AMI, smart meters, weather forecasts from weather services, and data from market or operational databases before passing through the same preprocessing pipeline described above. As such, adaptive filtering and interpolation are first applied for missing values, outliers, communication errors, and abnormal readings of the sensors. Then, the cleaned time series serve as the input to the MMPF–NARX forecasting layer, whose outputs can support day-ahead scheduling, reserve planning, demand-response activation, and maintenance planning.

From an operational standpoint, the framework proposed can be integrated into existing energy management systems (EMSs), distribution management systems (DMSs), or microgrid supervisory platforms. Specifically, the MMPF-NARX module would run periodically—for example, every hour or every day, depending on the forecasting horizon—and deliver updated load forecasts alongside error indicators and confidence diagnostics. Due to the modular architecture suggested, it is important to note that the operation does not need to supplant any existing utility tools with a new forecasting engine. As a result, the data can be aggregated to existing platforms and control-room dashboards through standard data interfaces as an analytics service. Finally, this is one of the main technical contributions provided, as it makes the framework realistic for gradual adoption in utility environments, where forecasting tools are typically designed to assist human operators and scheduling algorithms rather than to execute control actions directly.

The architecture also supports computational efficiency. NARX submodels run in parallel, and the final forecast is produced by adaptively weighing their outputs. With

the parallel processing resources available, adding more submodels does not necessarily increase execution time proportionally. As such, the optimization problem solved by either GARA or PSO is also low-dimensional, since each candidate solution represents only a weight vector over the available submodels. Analytically, our experimental results show that configurations of seven to ten submodels produced high accuracy without requiring an excessively large ensemble.

In addition, the comparison between GARA and PSO is also relevant for real-world use. PSO showed smoother convergence and lower computational effort than GARA. This is important in operational environments where forecasts must be updated continuously and computational resources are shared with other monitoring and control applications; this matter affects all processes and operations. For instance, GARA applies selection, crossover, mutation, and population sorting functions. Contrarily, PSO utilizes only a vector-based velocity update function and a position update function. For these reasons, one can argue that PSO is a more suitable forecasting technique for repeat and near-real-time cycles, whereas GARA can still be useful when there are no constraints on execution time and wider explorations of the weight space are beneficial.

Lastly, another important factor to discuss is how scalability can be achieved through spatial and hierarchical decomposition. In large power systems, the framework does not need to be trained as a single model covering the entire grid. Instead, separate MMPF–NARX modules can be assigned to individual feeders, substations, areas, customer clusters, or microgrids, with their outputs aggregated at a higher operational level for regional or system-wide forecasting. As such, our approach fits naturally with power network topology and supports distributed computing across servers, edge devices, and cloud infrastructure. Specifically, since the method uses multiple parallel submodels, additional computational resources can be allocated selectively to areas with greater variability, higher renewable penetration, or less reliable data.

4.5. Error Scatter Analysis and Symmetric Distribution Characteristics

The error scatter plots (Figures 10 and 12–21) provide a deeper insight into the nature of the residuals produced by each optimization method. The differences observed include the following.

1. *PSO precision and concentration:*
    - The PSO error scatter plots (Figures 10 and 12–15) are tightly grouped around.
    - This symmetric spread is smaller compared to GARA.
    - PSO particles tend to move, or "*swarm*", quickly toward the global best solution.
    - This is happening because the particles follow their own experience as well as the group's.
    - This swarm movement keeps forecasts close to each other and reduces the variation in errors.
2. *GARA Balance Diversity:*
    - The GARA error scatter plots (Figures 16–21) show a wider symmetric spread (greater variance).
    - They appear more symmetrical and balanced atop the horizontal axis than the PSO ones.
    - The mutation process creates diversity in the genetic algorithm and prevents its collapse into one solution.
    - As a result, the errors are more uniform or are even symmetrically spread.
    - Generally, they have bigger absolute values than the corresponding PSO ones, though over time they are less systematically biased.

- Using this feature can help avoid cumulative bias in longer-term operational horizons.

The differences previously mentioned can be explained by the bias–variance trade-off [48–50]. Due to the higher concentration of PSO on exploitation, its errors are fewer and forecasting is more accurate [51]. On the other hand, GARA focuses on exploration and diversity of the population via crossover and mutation. Due to this, this forecasting makes more balanced errors but is not as accurate as PSO.

For mission-critical applications, this difference is essential [52]. Also, the difference is critical for applications that we cannot afford to have a malfunction. As a result, tighter PSO error scatter is more suitable when the main objective is to minimize absolute forecast errors [53]. This means that at longer operational horizons, where cumulative bias ought to be avoided [54], the more symmetrical characteristic of the GARA residual distribution may be valuable.

4.6. Optimization Dynamics and Convergence Behavior

Another important factor to consider is the property characteristics of GARA and PSO algorithms as presented in Figures 22–27, i.e., the illustrations of our "fitness training error per iteration" curves. These illustrations provide insights into the internal dynamics of each optimization strategy. They show that, initially, the error is reduced fast; the final MAPE values are almost identical by the 40th iteration, and thus they perform very well [55]. As such, these results act as validation for our test cases, whereby both approaches employed can identify the best possible weight symmetric distribution for the hybrid MMPF-NARX framework.

Moreover, in terms of the shared initialization based on the MMPF's posterior probabilities, all the methods start early on from a similar error baseline. However, there is a distinct difference in their convergence path curves during the intermediate iterations.

1. *PSO Smoothness and Stability*
   - Smoother and more stable convergence curves toward the global minimum.
   - The error decreases more consistently.
   - PSO updates particles using velocity and momentum [56], as described earlier.
   - Each particle moves toward its own best and global best position, causing the swarm to move smoothly.
   - This prevents considerable fluctuations of the pattern.
2. *GARA Oscillatory*
   - Between the 10th generation and 30th generation, the path is more oscillatory.
   - The peaks indicate that some errors rise sharply on occasion but fall afterwards.
   - The genetic algorithm possesses a stochastic nature for this reason [57].
   - The addition of a mutation operator assures diversity and avoids convergence to any one region for any long period of time, which helps in exploring better solutions in the search space. Also, it creates spikes in performance during training.

The distinction in the convergence curves is the classic exploration–exploitation trade-off [57]. GARA's spikes indicate aggressive exploration, which sacrifices current accuracy in favor of broadening the search across a wider region of the parameter space. Conversely, PSO aims for an equitable exploitation of the available information, causing smoother behavior [58–60].

4.7. Computational Burden

The computational burden was assessed by observing the error trajectories in Figures 22–27. Although both algorithms were executed for 30 iterations, the effective

convergence (i.e., the point where $\Delta\text{Fitness} < 10^{-4}$) occurred significantly earlier for PSO. Mathematically, this is due to the velocity update rule (Equation (15)), which acts as a first-order momentum filter, accelerating the search toward the global optimum. In contrast, GARA's computational burden is increased by the selection operator, which requires sorting the entire population P in every generation, adding an $O(P\log P)$ overhead that becomes significant in real-time forecasting cycles. These computational differences are summarized in Table 3, which compares GARA and PSO in terms of complexity per iteration, convergence speed, FLOPs, memory overhead, and stability.

Table 3. Computational burden and convergence efficiency.

Metric	GARA	PSO	Improvement (%)
Complexity per Iteration	$O(P\log P + P \cdot L)$	$O(P \cdot L)$	~25% Faster
Avg. Iterations to Convergence	26–30	14–18	~45% Faster
Floating Point Ops (FLOPs)	High (Sorting/Randomizing)	Low (Vector Addition)	-
Memory Overhead	Moderate (Population Buffer)	Minimal (Velocity Vectors)	-
Stability (Figures 25–27)	Stochastic (High Variance)	Deterministic (Low Var.)	-

For GARA, the cost per iteration is higher due to the sorting and stochastic sampling:

$$C_{\text{GARA}} \approx N_{pop} \times (C_{\text{eval}} + C_{\text{select}} + C_{\text{cross}} + C_{\text{mut}}) \tag{20}$$

where C_{select} involves sorting the population, typically given as:

$$O(P\log P) \tag{21}$$

For PSO, the cost is lower because it only involves the vector arithmetic:

$$C_{\text{PSO}} \approx N_{pop} \times \left(C_{\text{eval}} + C_{\text{vel_update}} + C_{\text{pos_update}}\right) \tag{22}$$

where velocity/position updates are simple $O(1)$ vector additions.

Consequently, the fact that both methods reach almost the same final training error validates the robustness of the MMPF-NARX structure. Due to the lower computational load, PSO's smoother behavior may be desired for real-time energy management systems. On the flip side, GARA continues to be effective for the long-term allocation of resources.

5. Conclusions

This article presents a framework for simulating and modeling power load forecasting under uncertain conditions with symmetric and noisy data. A Multi-Model Partitioning Filter is combined with NARX submodels, with GARA and PSO used as adaptive optimization strategies to determine the final contribution weights of the parallel predictors.

Our work aligns with simulations and modeling in engineering, as it evaluates the forecasting framework under perturbation experiments. To approximate realistic measurement uncertainty, sensor error, and imperfect grid conditions, the test data were corrupted with correlated Gaussian noise. Cyclic temporal variables, such as hours and months, were encoded using unit-circle representations, and forecast residual symmetry and asymmetry were analyzed through error scatter plots.

The results showed that both GARA and PSO improved the robustness of the MMPF–NARX framework. PSO performed better overall across most configurations, achieving lower MAPEs, smoother convergence curves, and lower computational costs, making it

well-suited to real-time or mission-critical energy applications. GARA remains valuable as a strong evolutionary method, particularly where broader exploration of the weight space is beneficial.

Another interesting finding is that forecasting accuracy did not improve simply by increasing the number of NARX submodels. Nine submodels produced the best performance, and adding more increased model complexity without meaningful benefit. In some cases, additional submodels actually generalized worse under strong noise. This suggests the framework must balance model diversity against over-parameterization.

In summary, symmetry-aware modeling, adaptive optimization, and noise-based simulation provide a reliable foundation for electric load forecasting in modern power-system engineering. Future work should test the framework on larger datasets, demand symmetric profiles incorporating renewable energy, and real-time smart grid deployments, and it should examine more explicitly the relationship between residual symmetry, forecasting bias, and long-term operational stability. As such, the results suggest that the proposed system is an acceptable solution for application usage as a predictive layer to aid in decision-making from an engineering implementation point of view. Specifically, the optimized configuration of the MMPF-NARX framework by means of PSO can be used in practice for grid operation when frequent forecast updates are required, with lower computational burdens and stability in convergence. More specifically, this implementation should be considered in the case of day-ahead scheduling reserve estimation, demand-response planning, and operator situational awareness. Moreover, the GARA-based configuration remains advantageous in scenarios that require the exploration of an expansive region of the model-weight space. Similarly, offline planning cases or longer-horizon studies, according to our findings, showcase that merely increasing the number of NARX submodels is not guaranteed to yield improved results. Instead, the choice of model complexity should be made after validation in a realistic noise situation. Notably, in these present experiments, nine submodels offered the best compromise.

Lastly, as for future work, our framework provides a top-down approach for engineers in the field to build on. Firstly, we provide a broad framework to account for renewable-energy-related load and generation profiles, such as photovoltaic production, wind generation, net-load variability, and weather-related demand patterns, and investigate if the residual symmetry remains stable as renewable penetration tightens. Secondly, the framework should be put through real deployments on the grid or testing through near-real-time scenarios that involve continuous processing of smart meter, SCADA, AMI, and weather data streams in real-time now that the academics have been resolved. Lastly, intended verification in the future should analyze the frequency of updates and delays in computing, the intervals for retraining, the delays in communication, and the workings of fail-safes. Based on the above, these approaches of our framework will give researchers and practitioners a clearer picture of how the proposed forecasting approach performs under realistic circumstances.

Author Contributions: Conceptualization, S.P.; methodology, S.P.; software, S.P. and A.G.; validation, S.P. and N.E.M.; formal analysis, S.P.; investigation, S.P.; data curation, S.P.; writing-original draft preparation, S.P.; writing—review and editing, S.P., A.G. and N.E.M.; visualization, S.P.; supervision, N.E.M.; project administration, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw datasets used in this study are publicly available via Macrotrends (https://www.macrotrends.net/datasets/global-metrics/countries/grc/greece/gdp-gross-domestic-product, accessed on 1 March 2026), the World Bank (https://data.worldbank.org/

indicator/EG.USE.ELEC.KH.PC?locations=GR, accessed on 1 March 2026), the International Energy Agency (IEA, https://www.iea.org/countries/greece/electricity, accessed on 1 March 2026), and the Hellenic National Meteorological Service (EMY, https://emy.gr/en/announcements/675, accessed on 1 March 2026). The specific datasets analyzed during the current study are available from the corresponding author upon reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used DeepL Translator (version 26.2.1; DeepL SE, Cologne, Germany) as a neural machine translation tool for Greek-to-English translation and linguistic refinement of author-prepared text. In addition, during the preparation of this manuscript, the authors used Grammarly (version 1.2.210.1786; Grammarly, Inc.) for English-language editing, grammar checking, spelling correction, and readability improvement of the manuscript. These tools were used solely to improve the language and presentation of the manuscript. All scientific content, data analysis, interpretation, and conclusions were developed exclusively by the authors, who take full responsibility for the final content of the manuscript.

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

Appendix A

Appendix A.1. Derivation of MMPF Components

To align the theoretical framework with the physical problem of load forecasting, the following quantities are defined for a specific model θ as a continuation of the Multi-Model Partitioning Filter (MMPF) formulation, as presented in Equations (A1)–(A3).

The **Innovation Covariance** P_y is calculated as:

$$P_y(k|k-1;\theta) = H^T(k|\theta)P(k|k-1;\theta)H(k|\theta) + R(k) \tag{A1}$$

where $R(k)$ denotes the measurement noise covariance.

The **Innovation (Residual)** $\widetilde{y}$ represents the prediction error:

$$\widetilde{y}(k|k-1;\theta) = y(k) - H^T(k|\theta)\mathrm{B}(k+1,\ k|\theta)\hat{x}(k|k-1;\theta) \tag{A2}$$

where $B\ (\cdot)$ is the state transition matrix.

The **Observation Vector** H^T for the NARX structure is given by:

$$H^T(k|\theta) = [y(k-1), \ldots, y(k-p), e(k-1|\theta), \ldots, e(k-q|\theta)] \tag{A3}$$

where $e(k|\theta)$ is the local prediction error:

$$e(k|\theta) = y(k) - H^T(k|\theta)\hat{x}(k|k-1;\theta) \tag{A4}$$

Appendix A.2. NARX Submodel Configuration and Implementation Details

Unlike linear ARMA models, the NARX models employed as nonlinear predictors within the MMPF framework follow a *Polynomial NARX*, incorporating higher-order terms of the past load values and residuals:

$$y(k) = \sum a_i y(k-i) + \sum b_j e(k-j) + \sum\sum c_{i,j} y(k-i)y(k-j) + \ldots + u(k) \tag{A5}$$

a_i represents the autoregressive coefficients.

b_j represents the moving average coefficients.

$c_{i,j}$ represents the nonlinear coefficients, which enable the model to capture complex load dynamics.

$u(k)$ is the exogenous input vector.

The model order (p,q) is determined via the Akaike Information Criterion (AIC), which focuses on preventing overfitting and minimizing computational latency according to the Principle of Parsimony. The recursive estimation of the parameter vector $[a, b, c]$ is performed by the Extended Kalman Filter (EKF) through continuous linearization, ensuring its adaptation to load variations.

Table A1 below summarizes the main research streams identified in the literature review, their application focuses, their remaining limitations, and the way in which the proposed MMPF–NARX framework addresses these gaps. This table is included to clarify the research motivation of our study and to show that the proposed framework is positioned not only as another hybrid forecasting model but as a symmetry-aware, noise-robust, and complexity-controlled forecasting architecture.

Table A1. Summary of related forecasting streams and research gaps motivating the proposed framework.

Research Stream	Representative Approaches	Main Application Focus	Remaining Limitations/Gaps	How This Study Responds
Classical and hybrid time-series forecasting	ARIMA, ARIMA–PSO, regression-based hybrid models	Baseline and short-/medium-term load forecasting	Limited ability to represent nonlinear demand patterns and cyclic temporal symmetry	Uses nonlinear NARX submodels and unit-circle encoding for cyclic hour/month variables
Machine-learning and deep-learning forecasting	Random forests, gradient boosting, RNN/LSTM, ANN, SVM	Improved prediction accuracy under nonlinear load behavior	Often emphasizes point-forecast accuracy without examining residual symmetry or error-distribution behavior	Evaluates both MAPE and residual scatter behavior under moderate and heavy noise conditions
Probabilistic and ensemble forecasting	Ensemble models, quantile regression, kernel density estimation	Risk-aware forecasting and uncertainty representation	Distributional outputs are useful, but residual symmetry/asymmetry is not commonly used as a diagnostic criterion	Treats residual symmetry and asymmetry as diagnostic signals for comparing GARA and PSO behavior
Adaptive filtering and multi-model forecasting	MMPF, EKF-based adaptive models, multi-model filtering	Dynamic model adaptation and weighted prediction	Limited integration with nonlinear NARX submodels and limited analysis of temporal symmetry preservation	Combines MMPF with parallel NARX submodels and adaptive posterior-based weighting
GA/PSO-based optimization	GA, PSO, PSO–ANN, PSO–SVM, GA-based weight tuning	Parameter tuning, feature selection, and adaptive weight optimization	Few studies compare GA/PSO under the same MMPF–NARX structure while also testing noise robustness and model-complexity effects	Compares GARA and PSO under identical MMPF–NARX settings, moderate/heavy correlated noise, and 7–10 submodel configurations

This paper proposes a forecasting scheme that combines the MMPF and NARX models, as summarized in Figure A1 below.

Lastly, in Table A2, we present a comparative analysis of various case studies, indicating their approaches, the main design principles behind their design, and the strengths and weaknesses that were studied.

Table A2. Comparison based on approaches, main contributions, strengths, and limitations of recent research.

Refs.	Approaches	Main Concept	Strengths	Limitations/Gaps
[1,2]	ARIMA/ARIMA + PSO	Linear time-series forecasting with parameter tuning	Clear structure, easy implementation	Limited performance under nonlinear demand
[8]	Random forests, gradient boosting, RNN	Machine learning for nonlinear load patterns	Superior to classical ARIMA	Higher computational cost

Table A2. *Cont.*

Refs.	Approaches	Main Concept	Strengths	Limitations/Gaps
[9]	Ensemble probabilistic forecasting	Combination of multiple forecasts	Improved scalability and precision	Model management complexity
[11]	Quantile regression + KDE	Full probability distribution modeling	Better risk evaluation	Requires reliable density estimation
[14]	Hybrid with economic indicators	Use of early warning and industrial metrics	Improved performance near turning points	Depending on external data quality
[16]	Conditional hidden semi-Markov models	State-duration-based load modeling	Captures dynamic operating states	Sensitive to preprocessing
[18]	Gaussian filtering of smart meter data	Noise reduction before training	Cleaner input data	Additional preprocessing step
[19–21]	PSO + ANN	Swarm optimization for neural networks	Higher short-term accuracy	Possible local stagnation
[21,22]	GA-based MMPF weight optimization	Evolutionary adaptive filtering	Strong global search	Oscillatory convergence behavior
[22–24]	PSO + SVM	Swarm-based SVM tuning	Improved generalization	Increased training time
[25]	PSO + gray/wavelet models	Hybrid nonlinear forecasting	Better nonlinear handling	Layered model complexity
[26]	PSO for long-term forecasting	Swarm tuning for extended horizons	Stable convergence	Parameter sensitivity
[27]	PSO in power-system parameter estimation	Optimization of system cost functions	Efficient high-dimensional search	Initialization sensitivity
[28]	PSO theoretical analysis	Exploration–exploitation balance	Simple structure	Trade-off not fully eliminated
[38,39]	Bias–variance analysis	Theoretical explanation of trade-off	Conceptual clarity	Does not prescribe the optimal method
[61]	ANN + regression + SVR (fuzzy logic)	Hybrid combination of statistical and AI predictors	Better than standalone models	Increased model complexity
[62]	Hybrid with intelligent optimization	Optimization for weight tuning and feature selection	Improved accuracy	Depends on optimizer efficiency

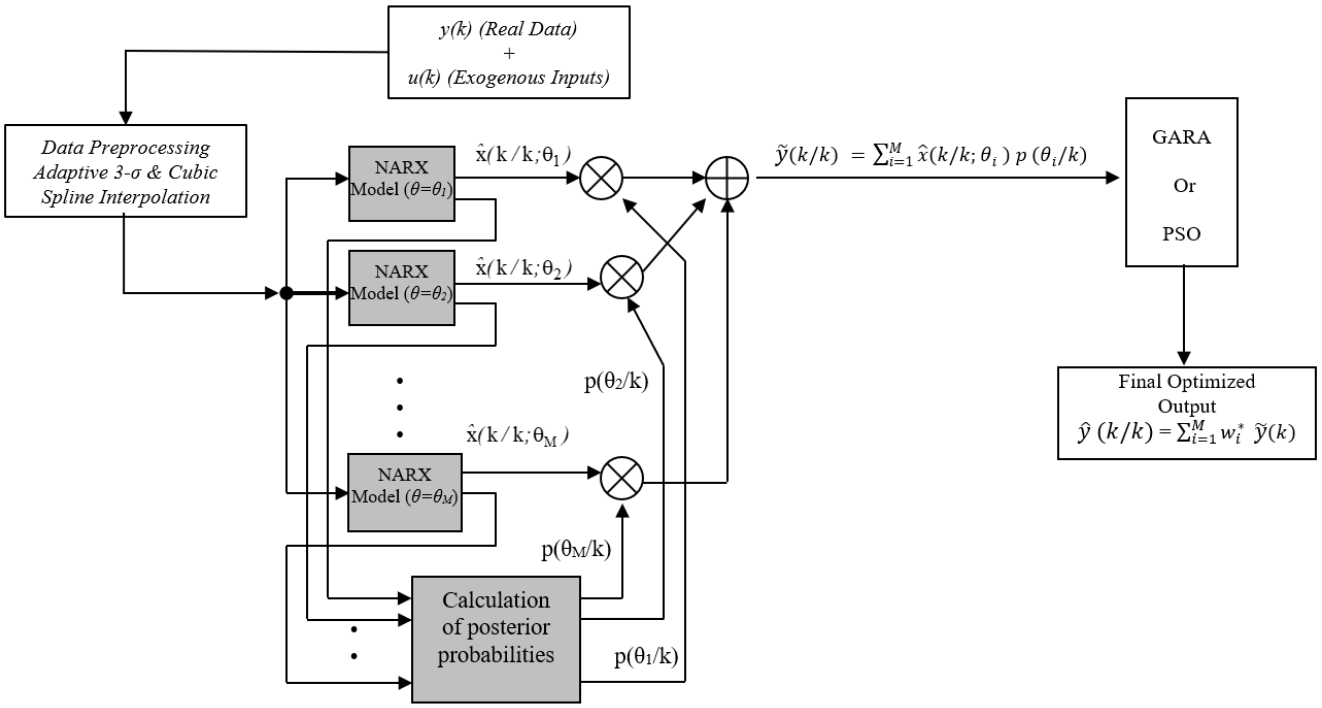


Figure A1. Hybrid MMPF–NARX forecasting architecture with GARA or PSO.

References

1. Marrero, L.; Garcia-Santander, L.; Carrizo, D.; Ulloa, F. An application of load forecasting based on ARIMA models and particle swarm optimization. In Proceedings of the 2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, Romania, 28–30 March 2019; pp. 1–6. [CrossRef]
2. Xiao, L.; Xiao, L. Combined modeling for electrical load forecasting with particle swarm optimization. In Proceedings of the 2014 IEEE Workshop on Electronics, Computer and Applications, Ottawa, ON, Canada, 8–9 May 2014; pp. 395–400. [CrossRef]
3. Shiblee, M.F.H.; Koukaras, P. Short-term load forecasting in the Greek power distribution system: A comparative study of gradient boosting and deep learning models. *Energies* **2025**, *18*, 5060. [CrossRef]
4. Gong, J.; Qu, Z.; Zhu, Z.; Xu, H.; Yang, Q. Ensemble models of TCN-LSTM-LightGBM based on ensemble learning methods for short-term electrical load forecasting. *Energy* **2025**, *318*, 134757. [CrossRef]
5. Dakheel, F.; Çevik, M. Optimizing smart grid load forecasting via a hybrid long short-term memory-XGBoost framework: Enhancing accuracy, robustness, and energy management. *Energies* **2025**, *18*, 2842. [CrossRef]
6. Li, K.; Yuan, L.; Qian, F.; Song, L.; Wu, X.; Wang, L.; Dai, J.; Shen, L. Short-Term Load Forecasting for Electricity Spot Markets Across Different Seasons Based on a Hybrid VMD-LSTM-Random Forest Model. *Energies* **2025**, *18*, 6097. [CrossRef]
7. Rani, S.; Ahmed, U.; Mahmood, A.; Manzoor, S.; Razzaq, S.; Jamil, U.; Zia, M.F.; Benbouzid, M. Dual Attention-Empowered Bidirectional Long Short-Term Memory Network for Short-Term Electric Load Forecasting. *Energy Technol.* **2026**, *14*, e202500914. [CrossRef]
8. Prashanthi, P.; Priyadarsini, K. A comparative study of the performance of machine learning based load forecasting methods. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore Nadu, India, 25–27 March 2021; pp. 132–136. [CrossRef]
9. Landgraf, A.J. An ensemble approach to GEFCom2017 probabilistic load forecasting. *Int. J. Forecast.* **2019**, *35*, 1432–1438. [CrossRef]
10. Gupta, T.; Bhatia, R.; Sharma, S. An ensemble-based enhanced short and medium term load forecasting using optimized missing value imputation. *Sci. Rep.* **2025**, *15*, 21857. [CrossRef] [PubMed]
11. Wang, S.; Wang, S.; Wang, D. Combined probability density model for medium term load forecasting based on quantile regression and kernel density estimation. *Energy Procedia* **2019**, *158*, 6446–6451. [CrossRef]
12. Liu, Y.; Mei, F.; Zhang, J.; Dai, X.; Li, W. Power Load Probabilistic Prediction Based on Multi-Value Quantile Regression and Timing Fusion Ensemble Learning Model. *Entropy* **2026**, *28*, 329. [CrossRef] [PubMed]
13. Shah, S.A.H.; Ahmed, U.; Bilal, M.; Khan, A.R.; Razzaq, S.; Aziz, I.; Mahmood, A. Improved electric load forecasting using quantile long short-term memory network with dual attention mechanism. *Energy Rep.* **2025**, *13*, 2343–2353. [CrossRef]
14. Wang, Y.; Wang, Z.; Li, X.; Wang, Z.; Liu, R.; Tong, X. Medium and long-term load intelligent forecasting method based on the comprehensive and multiple early warning indicators. *J. Phys. Conf. Ser.* **2020**, *1449*, 012124. [CrossRef]
15. Wang, X.; Wang, Z.; Yang, K.; Song, Z.; Bian, C.; Feng, J.; Deng, C. A survey on deep learning for cellular traffic prediction. *Intell. Comput.* **2024**, *3*, 0054. [CrossRef]
16. Ji, Y.; Buechler, E.; Rajagopal, R. Data-driven load modeling and forecasting of residential appliances. In Proceedings of the 2021 IEEE Power & Energy Society General Meeting (PESGM), Washington, DC, USA, 26–29 July 2021; p. 1. [CrossRef]
17. Wu, Z.; Wang, C.; Wu, J.; Wang, X.; Li, M.; Dong, Y.; Zhu, H.; Zhang, H. Dynamic adaptive modeling for non-intrusive load monitoring with unknown loads. *Energy Build.* **2025**, *329*, 115246. [CrossRef]
18. Rai, S.; De, M. Analysis of classical and machine learning based short-term and mid-term load forecasting for smart grid. *Int. J. Sustain. Energy* **2021**, *40*, 821–839. [CrossRef]
19. Abdullah, A.G.; Suranegara, G.M.; Hakim, D.L. Hybrid PSO-ANN Application for Improved Accuracy of Short Term Load Forecasting. *WSEAS Trans. POWER Syst.* **2014**, *9*, 446–451. Available online: https://wseas.com/journals/articles.php?id=5330 (accessed on 1 June 2026).
20. Shafiei Chafi, Z.; Afrakhte, H. Short-term load forecasting using neural network and particle swarm optimization (PSO) algorithm. *Math. Probl. Eng.* **2021**, *2021*, 5598267. [CrossRef]
21. Ozerdem, O.C.; Olaniyi, E.O.; Oyedotun, O.K. Short term load forecasting using particle swarm optimization neural network. *Procedia Comput. Sci.* **2017**, *120*, 382–393. [CrossRef]
22. Jiang, D. Study on short-term load forecasting method based on the PSO and SVM model. *Int. J. Control Autom.* **2015**, *8*, 181–188. [CrossRef]
23. Ji, G.; Li, S.; Shi, Z.; Zhang, X.; Zhao, W. Regional power load forecasting based on PSOSVM. In Proceedings of the 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 14–16 December 2018; pp. 1685–1688. [CrossRef]
24. Ren, G.; Wen, S.; Yan, Z.; Hu, R.; Zeng, Z.; Cao, Y. Power load forecasting based on support vector machine and particle swarm optimization. In Proceedings of the 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 12–15 June 2016; pp. 2003–2008. [CrossRef]

25. Bahrami, S.; Hooshmand, R.; Parastegari, M. Short term electric load forecasting by wavelet transform and grey model improved by PSO (particle swarm optimization) algorithm. *Energy* **2014**, *72*, 434–442. [CrossRef]
26. AlRashidi, M.; EL-Naggar, K. Long term electric load forecasting based on particle swarm optimization. *Appl. Energy* **2010**, *87*, 320–326. [CrossRef]
27. Alrashidi, M.R.; El-Naggar, K.M.; Al-Othman, A.K. Particle swarm optimization based approach for estimating the fuel-cost function parameters of thermal power plants with valve loading effects. *Electr. Power Compon. Syst.* **2009**, *37*, 1219–1230. [CrossRef]
28. Coello Coello, C.A.; Reyes-Sierra, M. Multi-objective particle swarm optimizers: A survey of the state-of-the-Art. *Int. J. Comput. Intell. Res.* **2006**, *2*, 287–308. [CrossRef]
29. Beligiannis, G.; Skarlas, L.; Likothanassis, S. A generic applied evolutionary hybrid technique. *IEEE Signal Process. Mag.* **2004**, *21*, 28–38. [CrossRef]
30. Katsikas, S.; Likothanassis, S.; Beligiannis, G.; Berkeris, K.; Fotakis, D. Genetically determined variable structure multiple model estimation. *IEEE Trans. Signal Process.* **2001**, *49*, 2253–2261. [CrossRef]
31. Mustaffa, Z.; Sulaiman, M.H. Advanced forecasting of building energy loads with XGBoost and metaheuristic algorithms integration. *Energy Storage Sav.* **2025**, *4*, 421–438. [CrossRef]
32. Camacho-Villalón, C.L.; Stützle, T.; Dorigo, M. Designing new metaheuristics: Manual versus automatic approaches. *Intell. Comput.* **2023**, *2*, 0048. [CrossRef]
33. Parvathareddy, S.; Yahya, A.; Amuhaya, L.; Samikannu, R.; Suglo, R.S. A hybrid machine learning and optimization framework for energy forecasting and management. *Results Eng.* **2025**, *26*, 105425. [CrossRef]
34. Saber, M.; Ouahi, M.; Motahhir, S.; El Akchioui, N. Neural Network-Aided Linear Parameter-Varying Modeling and Finite-Time Control for Autonomous Driving. *Intell. Comput.* **2026**, *5*, 0227. [CrossRef]
35. Hammam, I.M.; El-Kharbotly, A.K.; Sadek, Y.M. Adaptive demand forecasting framework with weighted ensemble of regression and machine learning models along life cycle variability. *Sci. Rep.* **2025**, *15*, 38482. [CrossRef] [PubMed]
36. Avinash, G.; Mishra, S. Bayesian model averaging based deep learning forecasts of inpatient bed occupancy in mental health facilities. *Sci. Rep.* **2025**, *15*, 38294. [CrossRef] [PubMed]
37. Pappas, S.; Ekonomou, L.; Karampelas, P.; Katsikas, S.; Liatsis, P. Modeling of the grounding resistance variation using ARMA models. *Simul. Model. Pract. Theory* **2008**, *16*, 560–570. [CrossRef]
38. Pappas, S.; Ekonomou, L.; Karamousantas, D.; Chatzarakis, G.; Katsikas, S.; Liatsis, P. Electricity demand loads modeling using autoregressive moving average (ARMA) models. *Energy* **2008**, *33*, 1353–1360. [CrossRef]
39. Pappas, S.S.; Karamousantas, D.C.; Chatzarakis, G.E.; Pappas, C.S. A New Hybrid Forecasting Strategy Applied to Mean Hourly Wind Speed Time Series. *J. Wind Energy* **2014**, *2014*, 683939. [CrossRef]
40. Pappas, S.S.; Ekonomou, L.; Moussas, V.C.; Karampelas, P.; Katsikas, S.K. Adaptive load forecasting of the Hellenic electric grid. *J. Zhejiang Univ. Sci. A* **2008**, *9*, 1724–1730. [CrossRef]
41. Pappas, S.S.; Adam, S. Prediction of the long-term electrical energy consumption in Greece using adaptive algorithms. *WSEAS Trans. Power Syst.* **2018**, *13*, 291–299.
42. Pappas, S. Application and comparison of evolutionary techniques for forecasting the Hellenic grid electricity load. *Int. J. Power Energy Res.* **2017**, *1*, 139–149. [CrossRef]
43. Sakib, M.; Siddiqui, T.; Mustajab, S.; Alotaibi, R.M.; Alshareef, N.M.; Khan, M.Z. An ensemble deep learning framework for energy demand forecasting using genetic algorithm-based feature selection. *PLoS ONE* **2025**, *20*, e0310465. [CrossRef] [PubMed]
44. Tsoulos, I.G.; Charilogis, V.; Tsalikakis, D. Introducing a new genetic operator based on differential evolution for the effective training of neural networks. *Computers* **2025**, *14*, 125. [CrossRef]
45. Montgomery, D.C. *Introduction to Statistical Quality Control*; John Wiley & Sons: Hoboken, NJ, USA, 2019. Available online: https://www.wiley.com/en-us/shop/general-introductory-industrial-engineering/introduction-to-statistical-quality-control-8th-edition-p-9781119399308 (accessed on 1 June 2026).
46. Zhang, Z.; Xu, W.; Gong, Q. Short-term power load forecasting based on particle swarm optimization long short-term memory neural network. In Proceedings of the 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 24–26 February 2023; pp. 412–416. [CrossRef]
47. Zhou, F.; Zhou, H.; Li, Z.; Zhao, K. Multi-step ahead short-term electricity load forecasting using VMD-TCN and error correction strategy. *Energies* **2022**, *15*, 5375. [CrossRef]
48. Valentini, G.; Dietterich, T. *Bias—Variance Analysis and Ensembles of SVM*; Lecture Notes in Computer Science Springer; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2364, pp. 221–231. [CrossRef] [PubMed]
49. Geman, S.; Bienenstock, E.; Doursat, R. Neural networks and the bias/Variance dilemma. *Neural Comput.* **1992**, *4*, 1–58. [CrossRef]
50. Hosseini, E.; Møller, J.K.; Banaei, M.; Ebrahimy, R. Optimal weights and Heuristic Meta-Models: Learning-based multi-model strategies for energy forecasting. *Array* **2025**, *28*, 100550. [CrossRef]
51. Wu, C.; Zhang, Z. Surfing information: The challenge of intelligent decision-making. *Intell. Comput.* **2023**, *2*, 0041. [CrossRef]

52. Wang, S.; Nikfar, M.; Agar, J.C.; Liu, Y. Stacked Deep Learning Models for Fast Approximations of Steady-State Navier–Stokes Equations for Low Re Flow. *Intell. Comput.* **2024**, *3*, 0093. [CrossRef]
53. Tindall, J.R.; Rogage, K.; Doukari, O. Optimising deep learning for smart building energy prediction: A particle swarm approach with real-world validation. *J. Build. Eng.* **2025**, *116*, 114652. [CrossRef]
54. Lengler, J.; Offermann, T. Diversity-Preserving Exploitation of Crossover. In Proceedings of the 18th ACM/SIGEVO Conference on Foundations of Genetic Algorithms, Leiden, The Netherlands, 27–29 August 2025; pp. 154–165. [CrossRef]
55. Khan, S.; Mazhar, T.; Shahzad, T.; Ali, T.; Ayaz, M.; Ghadi, Y.Y.; Aggoune, E.H.M.; Hamam, H. Optimizing load demand forecasting in educational buildings using quantum-inspired particle swarm optimization (QPSO) with recurrent neural networks (RNNs): A seasonal approach. *Sci. Rep.* **2025**, *15*, 19349. [CrossRef] [PubMed]
56. Yang, B.; Zhang, Q.L. Parallelizing a modified particle swarm optimizer (PSO). *Adv. Mater. Res.* **2010**, *163–167*, 2404–2409. [CrossRef]
57. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Boston, MA, USA, 1989; Available online: https://dl.acm.org/doi/10.5555/534133 (accessed on 1 June 2026).
58. Slowik, A.; Kwasnicka, H. Evolutionary algorithms and their applications to engineering problems. *Neural Comput. Appl.* **2020**, *32*, 12363–12379. [CrossRef]
59. Osaba, E.; Villar-Rodriguez, E.; Del Ser, J.; Nebro, A.J.; Molina, D.; LaTorre, A.; Ponnuthurai, N.S.; Coello, C.C.A.; Herrera, F. A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems. *Swarm Evol. Comput.* **2021**, *64*, 100888. [CrossRef]
60. Črepinšek, M.; Liu, S.; Mernik, M. Exploration and exploitation in evolutionary algorithms. *ACM Comput. Surv.* **2013**, *45*, 1–33. [CrossRef]
61. Keitsch, K.A.; Bruckner, T. Modular electrical demand forecasting framework—A novel hybrid model approach. In Proceedings of the 2016 13th International Multi-Conference on Systems, Signals & Devices (SSD), Leipzig, Germany, 21–24 March 2016; pp. 454–458. [CrossRef]
62. Ghiasi, M.; Irani Jam, M.; Teimourian, M.; Zarrabi, H.; Yousefi, N. A new prediction model of electricity load based on hybrid forecast engine. *Int. J. Ambient Energy* **2017**, *40*, 179–186. [CrossRef]

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