mathematics

MDPI

Article

Optimization of a Hybrid EKF-ANN Model via Double-Criterion Early Stop Pruning for Enhanced Wind Speed Forecasting

Athanasios Donas [1,2,*], George Galanis [2], Ioannis Pytharoulis [3] and Ioannis Th. Famelis [1]

[1] microSENSES Laboratory, Department of Electrical and Electronics Engineering, Polytechnic School, University of West Attica, Ancient Olive Grove Campus, 250, Thivon Ave., Egaleo, 12241 Athens, Greece; ifamelis@uniwa.gr
[2] Mathematical Modeling and Applications Laboratory, Hellenic Naval Academy, Hatzikiriakion, 18539 Piraeus, Greece; ggalanis@hna.gr
[3] Department of Meteorology and Climatology, School of Geology, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; pyth@geo.auth.gr
* Correspondence: adonas@uniwa.gr

Abstract

A novel double-criterion early stopping pruning strategy is introduced in this study. The proposed approach enables the structural optimization of a hybrid filtering framework that integrates FeedForward Neural Networks with an adaptive Extended Kalman Filter, by simultaneously monitoring the validation error and the trace of the error covariance matrix. Unlike classical pruning methods, which are applied after the completion of the training process and aggressively remove network neurons, the proposed scheme exploits the learning procedure, achieving a more selective reduction of 2% to 13%, balancing effectively between strong generalization performance and computationally efficient training. The proposed framework is evaluated on wind speed forecasts obtained from a numerical weather prediction model, within a time-varying window scheme, demonstrating promising improvements. Key statistical indices, such as the Mean Absolute Error and the Root Mean Square Error, were significantly reduced, with reductions ranging from approximately 65% to 80% and 60% to 78%, respectively. These findings suggest that the proposed methodology offers a robust and accurate framework for time series forecasting in operational settings.

Keywords: Extended Kalman Filters; FeedForward Neural Networks; hybrid algorithms; overfitting; pruning

MSC: 62M20; 93E11; 68T07

Check for updates

Academic Editors: Fan Wang and Qiang Li

Received: 19 April 2026
Revised: 5 May 2026
Accepted: 10 May 2026
Published: 13 May 2026

1. Introduction

Accurate forecasting of time series generated by stochastic dynamical systems remains a challenging problem in applied mathematics, particularly in the presence of uncertainty and model imperfections. In many real-world applications, the underlying processes are governed by complex, nonlinear dynamics, rendering purely deterministic or purely data-driven approaches insufficient for achieving reliable predictions. As a result, methodologies that utilize heavy statistical estimation techniques or advanced machine learning models have attracted increasing attention.

Kariniotakis and Pinson [1] demonstrated the effectiveness of fuzzy logic models in managing the inherent uncertainty of wind power data, while Kariniotakis et al. [2]

established that the performance of such predictive models is heavily influenced by site-specific environmental characteristics. The importance of long-term stochastic modeling in capturing the variability of extreme physical phenomena was further explored by Vanem [3], providing a rigorous framework for time-dependent environmental variables. From a systemic perspective, Giebel [4] emphasized the benefits of understanding distributed power generation in mitigating the variability of stochastic energy sources. Furthermore, the mathematical foundations of risk and uncertainty in such complex systems were addressed by Resconi [5] through the geometry of morphogenetic systems.

Recent advances in time-series forecasting highlight the effectiveness of hybrid and error-correction-based approaches. Chang et al. [6] proposed a hybrid framework combining subsequence correction with multi-scale deep learning regression for long-term wind power forecasting, demonstrating improved performance in stochastic energy systems. Similarly, Aminjan et al. [7] addressed non-stationary and multi-regime dynamics in complex physical processes such as gas–liquid two-phase flows, where deep learning models are used to capture regime transitions and abrupt changes in system behavior.

Motivated by these requirements, hybrid estimation frameworks that integrate state-space modeling with neural network-based function approximation offer a promising direction for robust forecasting under uncertainty. Donas et al. [8] introduced such a hybrid framework that effectively enhances numerical weather and wave prediction models by dynamically adjusting to non-linear error patterns through parametrized ANNs. Within this context, the Hybrid Extended Kalman Filter (HEKF), which combines FeedForward Neural Networks (FFNNs) with an adaptive Extended Kalman Filter (EKF), has been successfully employed to produce accurate time series forecasts, reducing both systematic and non-systematic components of forecast error.

In this framework, an FFNN is first utilized to estimate an exogenous parameter (parametrized FeedForward Neural Network-pFFNN) governing the evolution of the EKF covariance matrices. Afterwards, based on the estimated parameter, the EKF is applied as a sequential training algorithm for a second FFNN, which generates the final improved predictions. This two-stage structure ensures both adaptability and computational efficiency, making the framework suitable for practical implementations.

Despite its effectiveness, the HEKF scheme presents a notable limitation, since the phenomenon of overfitting is poorly addressed, specifically in its second stage. Overfitting occurs when a model learns the training data too closely, resulting in reduced generalization performance on unseen data. Consequently, as Jabbar and Khan [9] note, controlling model complexity through robust strategies is crucial for maintaining predictive reliability in supervised learning tasks and for mitigating overfitting. Therefore, the present study extends the original HEKF formulation by introducing a novel double-criterion early stopping and pruning strategy to enhance its predictive performance.

More precisely, the proposed approach integrates validation-driven learning with a covariance-informed structural pruning mechanism, enabling the simultaneous control of model complexity and estimation uncertainty. Specifically, the training process is guided by the joint minimization of the validation error and the trace of the error covariance matrix. In addition, pruning decisions are based on the asymptotic properties of the error covariance matrix, allowing the removal of entire connection paths with negligible contribution. The novelty of the suggested modification lies in embedding the pruning mechanism within the sequential learning dynamics of the EKF, in contrast to classical pruning approaches that are applied after the training procedure. This formulation provides a statistically consistent framework for mitigating overfitting in hybrid filtering systems.

The effectiveness of the proposed mechanism is evaluated through its application to wind speed forecasts derived from the Weather Research and Forecasting (WRF) numerical weather prediction model with the Advanced Research dynamic core. More thoroughly, the analysis is conducted within a time-window framework, allowing for sequential training and evaluation. The suggested hybrid scheme is compared against the baseline HEKF, classical pruning techniques, and the raw WRF outputs, in terms of forecasting accuracy and computational efficiency.

In summary, the novelty and key contributions of the proposed study are as follows:

- Novel Double-Criterion Strategy: Introduction of a data-driven early stopping pruning mechanism that monitors both the validation error and the trace of the error covariance matrix simultaneously.
- Dynamic Structural Optimization: Pruning occurs dynamically during the EKF learning process, ensuring real-time model refinement rather than post-training adjustment.
- Selective Complexity Reduction: Achieves a balanced network by removing only 2–13% of redundant neurons, preserving essential non-linear mapping capabilities.
- Performance and Efficiency: Significant reduction in errors (up to 80% in MAE) combined with a 50% decrease in computational training time.

2. Methodology

The proposed hybrid framework is introduced in this section. Specifically, Section 2.1 presents the overall architecture, while Section 2.2 presents the core components of the HEKF. Section 2.3 introduces the proposed double-criterion pruning methodology, whereas Section 2.4 outlines the WRF time-window experimental design.

2.1. Hybrid Framework Overview

The original HEKF is structured into two main stages: (i) a selection stage, in which a FeedForward Neural Network is trained across different configurations of hidden neurons and memory parameter values to define the optimal combination, and (ii) an adaptive Extended Kalman Filter stage, acting as a training algorithm for a second FeedForward Neural Network, which produces the improved forecasts for the wind speed.

Initially, the available data, consisting of WRF forecasts and corresponding (recorded) observations, were partitioned into two subsets: the Training Dataset (TrDs) and the Testing Dataset (TesDs). The TrDs dataset trains the pFFNN and identifies the optimal combination, acting as a parameter estimation mechanism. To achieve this, the TrDs was further divided into the training data set (trds) and the validation data set (valds), where the former was employed to update the network weights, and the latter was used to monitor its (validation) error and determine the stopping point of the training process. That concludes the first stage of the initial HEKF approach.

Subsequently, the adaptive EKF framework was employed to train a distinct feed-forward network, utilizing the optimal configuration obtained from the parametrized FFNN. More specifically, the chosen memory parameter updates dynamically the state and measurement covariance matrices during the EKF algorithm, while the selected number of hidden neurons determines the size of the second FeedForward Neural Network.

At this point of the second stage, the proposed expansion was introduced. In contrast to the baseline HEKF, which processes the TrDs in a single pass to estimate the optimal weights, the proposed framework suggests an iterative strategy. Particularly, the already defined training and validation subsets (trds and valds) were exploited over multiple passes, simulating the epochs of a conventional artificial neural network training process. During each epoch, the weights that minimize the (validation) error and also produce the lowest trace of the error covariance matrix were retained. Here, an epoch is defined as a complete pass through the entire training dataset (trds), during which all input–output pairs were sequentially processed.

Based on this dual-criterion selection, the pruning process was carried out by evaluating the contribution of each connectivity path to the network's performance. The importance of each path was quantified using second-order information from the EKF error covariance matrix, capturing the influence of each path on the network's predictive accuracy as measured on the validation dataset. If the aggregated contribution of a given path fell below a predefined threshold, the entire connection associated with the corresponding neuron was pruned.

The reduced network, along with the corresponding weights and the error covariance matrix, was used to initialize the subsequent epoch. In this way, the training of the second FFNN was no longer based on a single-pass estimation, but on a validation-driven strategy that boosts its generalization capability and effectively mitigates overfitting effects. If the error was not reduced for several epochs, particularly ten, the training process was terminated, using the weights that satisfy the double-criterion check. Thus, an Early Stop pruning was established for the adaptive Extended Kalman Filter.

Once the training process of this hybrid filter was completed, the optimized (second) network architecture, using the TesDs dataset, was employed to generate improved forecasts for the environmental parameter under study. These were compared, in terms of predictive accuracy and computational efficiency, with those derived from the initial methodology and the classical pruning EKF technique, in which the pruning process was performed after training completion based on the final error covariance matrix.

Overall, the proposed framework consists of two sequential stages involving two distinct neural networks. The first network was employed to determine the optimal architecture and memory parameter, while the second network was trained using the EKF, incorporating the information derived from the first stage. This separation ensures a clear distinction between structural optimization and adaptive learning. More precisely, in the first stage, an FFNN was trained by minimizing the Mean Squared Error, which constitutes the objective function of this stage. In the second stage, the training process of a different FFNN was formulated as a recursive minimization of the forecast error, defined as the difference between observed values and model forecasts, which defines the objective function of the second stage. Therefore, each stage of the proposed hybrid framework was associated with a distinct objective function, operating in a complementary and sequential manner.

The key components of the proposed hybrid modification are illustrated in Figure 1.

2.2. Baseline HEKF Formulation

This sub-section presents the baseline Hybrid Extended Kalman Filter, highlighting its core components: (i) the parametrized FeedForward Neural Network and (ii) the adaptive Extended Kalman Filter.

2.2.1. Parametrized FeedForward Neural Networks

Artificial Neural Networks [10,11] are supervised machine learning mechanisms inspired by the functions of the human brain, aiming to unlock complex, non-linear relationships between independent input variables and their corresponding target outputs. ANNs are composed of interconnected processing units, named neurons, that communicate through weighted connections. Each neuron receives inputs and applies an activation function to generate a corresponding output, enabling information to propagate throughout the entire network. Therefore, a wide range of ANN architectures can be constructed [12].

https://doi.org/10.3390/math14101650

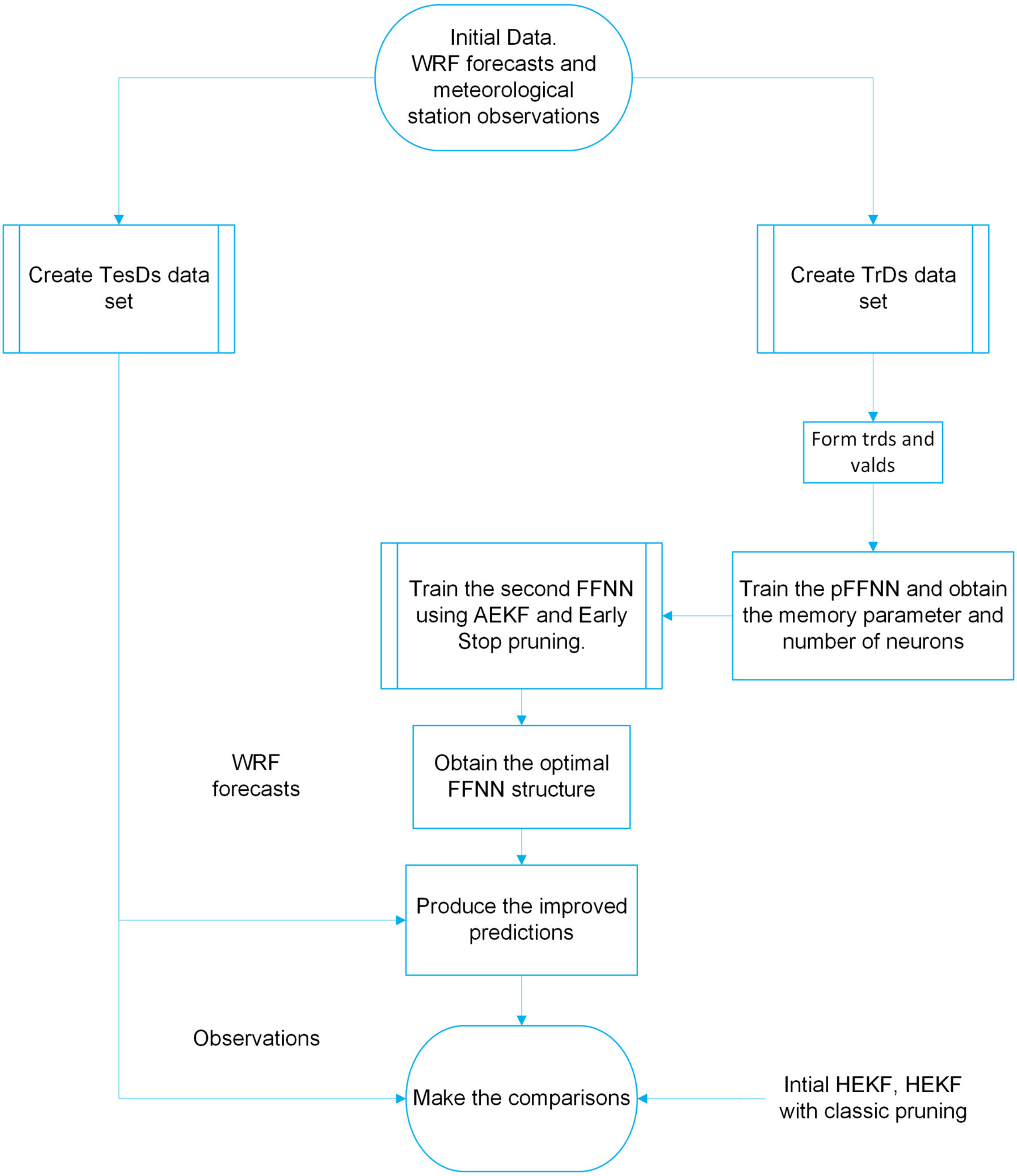

Figure 1. Hybrid Extended Kalman Filter. Method diagram.

This study, however, focuses on FeedForward Neural Networks [13]. FFNNs establish a mapping between input data and output responses through a unidirectional flow of information. They generally consist of multiple fully connected layers organized in a directed acyclic topology. Here, a fully connected, three-layer FFNN structure was employed, including a single hidden layer. According to the universal approximation theorem, it is sufficient to approximate nonlinear mappings with arbitrary accuracy. Furthermore, as an activation function, the suggested architecture utilizes the log-sigmoid [14–16]:

$$\varphi(x) = \frac{1}{1+e^{-x}}. \tag{1}$$

The proposed network was not designed to operate as a baseline forecasting tool, but rather as an estimation mechanism that determines the optimal combination of the number of hidden neurons and the value of a memory parameter, forming the so-called parametrized FeedForward Neural Network. To this end, the proposed network (Figure 2) utilizes an input matrix defined as:

$$X = (X_1, \ldots, X_t, \ldots, X_N,),$$

where N presents the size of the TrDs, and X_t represents the input vector at time t, consisting of two components: the WRF forecast (for_t) and the memory parameter (a); i.e.,

$$X_t = [for_t, a]^T.$$

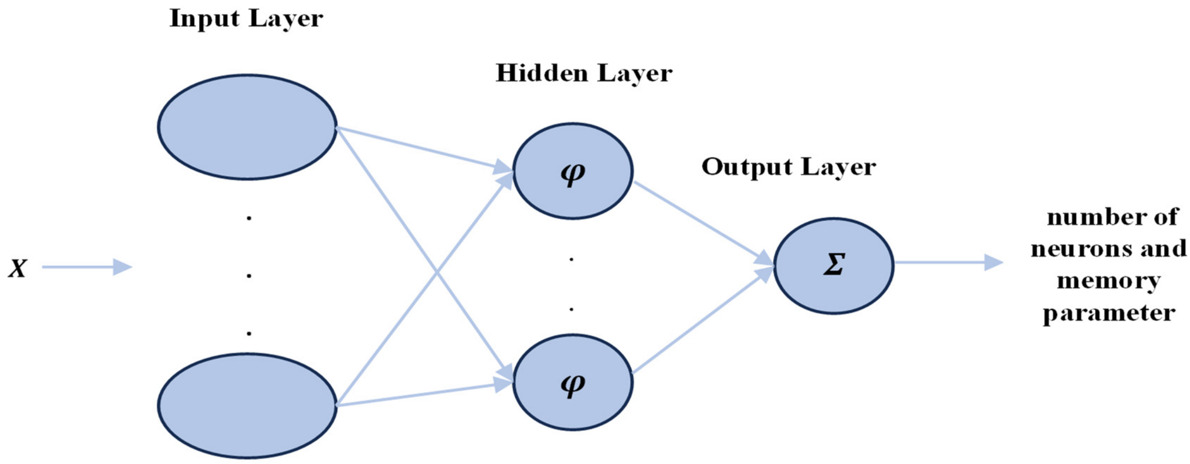

Figure 2. A three-layer parametrized FeedForward Neural Network.

Based on this formulation, the pFFNN was trained for multiple values of the parameter a, aiming to identify the one that best captures the relationship between the input vectors and the corresponding targets, and for multiple numbers of hidden neurons. The parameter was selected within the range $[0.3,\ 0.4]$, while the number of neurons varies between 20 and 30. The structure that minimizes the Mean Squared Error (MSE) was considered optimal:

$$MSE = \frac{1}{N}\sum_{t=1}^{N} \left(er^t\right)^2, \tag{2}$$

where $er^t = (T_t - X_t)$ denotes the training error between the scalar target T_t and the corresponding input vector X_t.

To minimize the MSE, an appropriate optimization procedure must be employed. Specifically, the Levenberg–Marquardt (LM) method was adopted in this study for training the pFFNN, due to its efficiency in problems involving a small number of the network's weights and also because the MSE is used as a performance index [17]. The Levenberg–Marquardt algorithm integrates the principles of the steepest descent and Gauss–Newton methods to iteratively adjust the network weights during training. More specifically, at iteration k, the weight update is defined as follows [18]:

$$W_{k+1} = W_k - \left(J_k^T J_k + \mu I\right) J_k er_k, \tag{3}$$

where

- W_k is a column vector consisting of the network's variables (weights and biases).
- J_k is the Jacobian matrix.
- er_k represents the vectorized training error, i.e., $er_k = \left[er_k^1, \ldots, er_k^t, \ldots er_k^N\right]^T$.
- The quantity $\left(J_k^T J_k + \mu I\right)$ denotes an approximation of the Hessian matrix introduced by the LM algorithm, i.e., $Hes = \left(J_k^T J_k + \mu I\right)$.
- I is the identity matrix.

- μ is a combination parameter that takes positive values, ensuring the reversibility of the $J_k^T J_k$ matrix.

The computation of the Jacobian matrix constitutes a key step in the implementation of the LM algorithm, since it can be computationally intensive depending on the complexity of the network structure (number of neurons and hidden layers). In the case of the proposed parametrized FFNN, the Jacobian matrix is expressed as follows:

$$J_k = \frac{2}{N}\begin{pmatrix} \frac{\partial er_k^1}{\partial d_{1,1}} & \frac{\partial er_k^1}{\partial d_{2,1}} & \cdots & \frac{\partial er_k^1}{\partial d_{1,M}} & \frac{\partial er_k^1}{\partial d_{2,M}} & \frac{\partial er_k^1}{\partial b_1} & \cdots & \frac{\partial er_k^1}{\partial b_M} & \frac{\partial er_k^1}{\partial v_1} & \cdots & \frac{\partial er_k^1}{\partial v_M} & \frac{\partial er_k^1}{\partial b_{out}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial er_k^t}{\partial d_{1,1}} & \frac{\partial er_k^t}{\partial d_{2,1}} & \cdots & \frac{\partial er_k^t}{\partial d_{1,M}} & \frac{\partial er_k^t}{\partial d_{2,M}} & \frac{\partial er_k^t}{\partial b_1} & \cdots & \frac{\partial er_k^t}{\partial b_M} & \frac{\partial er_k^t}{\partial v_1} & \cdots & \frac{\partial er_k^t}{\partial v_M} & \frac{\partial er_k^t}{\partial b_{out}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial er_k^N}{\partial d_{1,1}} & \frac{\partial er_k^N}{\partial d_{2,1}} & \cdots & \frac{\partial er_k^N}{\partial d_{1,M}} & \frac{\partial er_k^N}{\partial d_{2,M}} & \frac{\partial er_k^N}{\partial b_1} & \cdots & \frac{\partial er_k^N}{\partial b_M} & \frac{\partial er_k^N}{\partial v_1} & \cdots & \frac{\partial er_k^N}{\partial v_M} & \frac{\partial er_k^N}{\partial b_{out}} \end{pmatrix}$$

wherein this case, we have the following:

- $d_{s,l}$ express the synaptic weight between the s input and the l neuron in the hidden layer. Here, $s = 1, 2$ and $l = 1, \ldots, M$.
- b_l and v_l are the biases of the l neuron and the corresponding synaptic weight between this neuron and the output layer, respectively.
- $\frac{\partial er_k^t}{\partial d_{s,l}} = -v_l \varphi\prime(net_t^l) X_t^s$ and $\frac{\partial er_k^t}{\partial b_l} = v_1 \varphi\prime(net_t^l)$, with net_t^l being the input of the l neuron based on the t training pattern, i.e., $net_t^l = \sum_{s=1}^2 d_{s,l} X_t^s + b_l$, where X_t^s is the s input of the X_t input vector.
- $\frac{\partial er_k^t}{\partial v_l} = -v_l \varphi(net_t^l)$ and $\frac{\partial er_k^t}{\partial b_{out}} = -1$, where b_{out} is the bias of the output layer.

Once the Jacobian matrix is computed, the training process of the suggested pFFNN based on the LM algorithm is illustrated in Figure 3.

Apart from the training algorithm, another issue that emerges when dealing with ANNs is the phenomenon of overfitting [9]. To mitigate this malfunction and improve the network's generalization, an early stopping strategy is employed [19]. During training, the network weights are updated using the training dataset at each epoch, while its performance is simultaneously evaluated on a validation dataset. If the validation error increases for several epochs, the training process is terminated, and the set of weights corresponding to the minimum validation error is selected as optimal (Figure 4). Properly handling overfitting is crucial for the successful implementation of the pFFNN.

Summarizing the above process, Table 1 presents the main characteristics of the proposed parametrized network, while in Appendix A, a detailed analysis of the first stage of the Hybrid Extended Kalman Filter is provided.

Table 1. Basic elements of the parametrized FeedForward Neural Network.

ANN	FeedForward Neural Network
Neurons	20 to 30
Activation Function	Log-Sigmoid
Hidden Layers	1
Output Layer	One Linear Output
Memory Parameter a	0.3:0.01:0.4
Training Algorithm	Levenberg–Marquardt, $\mu = 0.01$
Overfitting	Early Stop

https://doi.org/10.3390/math14101650

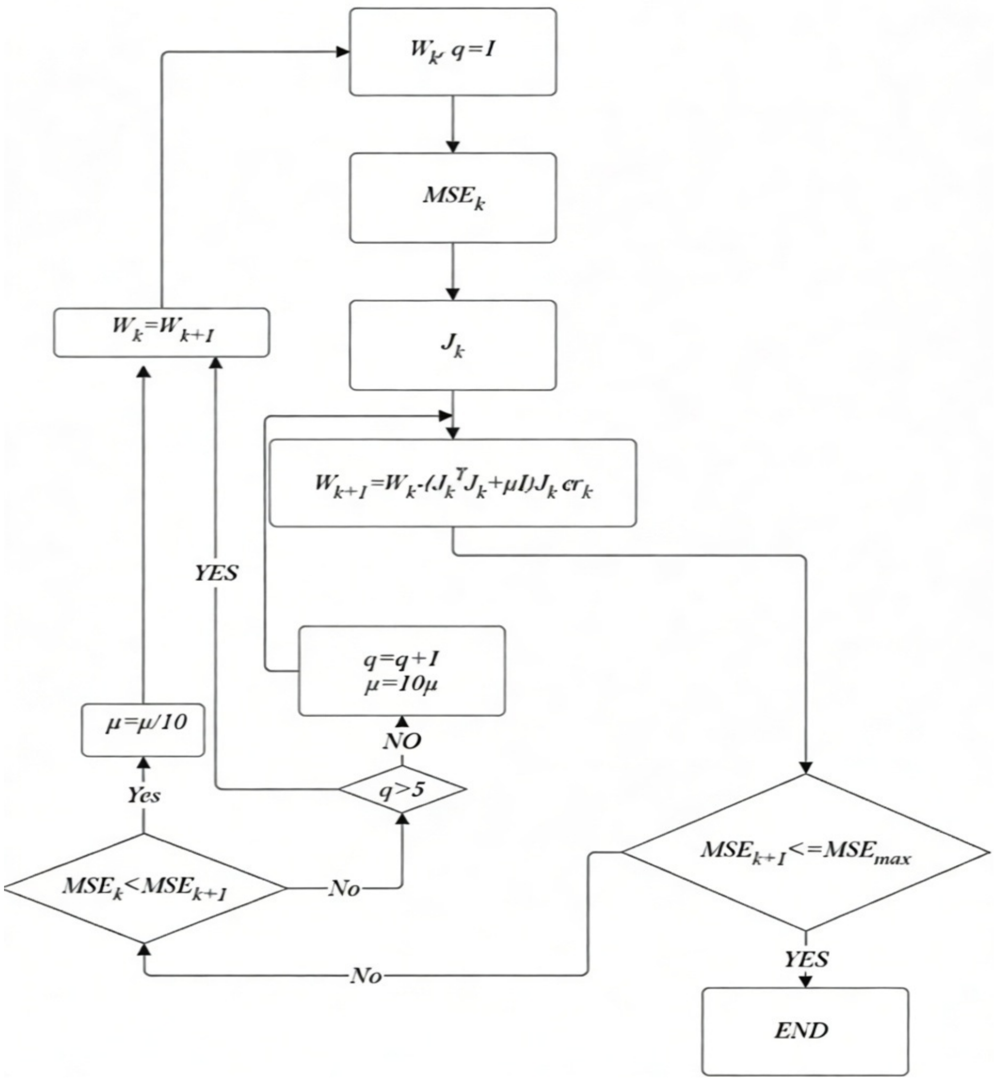

Figure 3. Levenberg–Marquardt training process for the pFFNN [8].

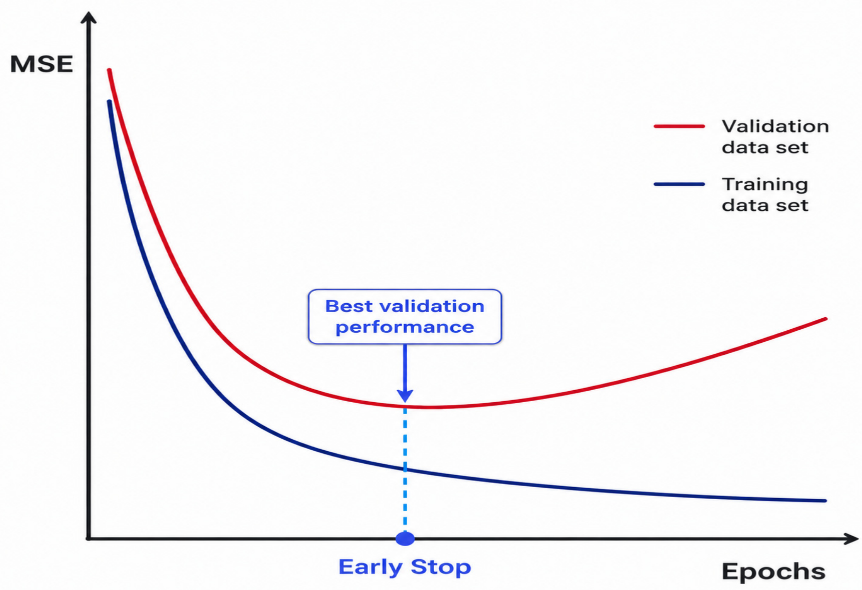

Figure 4. pFFNN. Early stopping strategy [6].

2.2.2. Adaptive Extended Kalman Filter

Once the training process of the pFFNN was finalized, the optimal combination of neurons and parameter a was utilized in the following way: (i) the number of neurons determines the size of a second, three-layer, and fully connected FeedForward Neural Network, comprising a single hidden layer with a log-sigmoid activation function, that is trained using the Extended Kalman Filter [20,21], whereas (ii) the memory parameter was applied to update the state and the measurement covariance matrices during the EKF algorithm.

The selected network structure was consistent with the first stage of the proposed hybrid filter and also with the EKF-based training framework, as deeper architectures would significantly increase the dimensionality of the state vector and the associated computational cost of the covariance matrix updates. Therefore, an adaptive Extended Kalman Filter was created and employed as a learning rule, intending to determine the network's optimal parameters.

Similar modified EKF algorithms have been frequently utilized as neural network training methods [22–25] due to their fast and efficient implementation [26]. These approaches are closely connected to sequential versions of the Gauss–Newton algorithm, and unlike the comparable batch variant, they do not require inversion of the approximate Hessian matrix [17].

To illustrate the training procedure of the adaptive EKF, a state-measurement model must be defined (Equations (4) and (5)). In this study, the corresponding model is formulated as:

$$W_t = W_{t-1} + w_{t-1} \tag{4}$$

$$y_t = f_t(W_t, for_t) + v_t, \tag{5}$$

where W_t is a column vector that concludes the total weights of the FFNN at time t. The quantity y_t presents the forecast error, defined as the difference between the recorded observation and the corresponding WRF forecast, at the same time. The variables w_{t-1} and v_t denote the Gaussian nonsystematic parts of the model error, with covariance matrices Q_{t-1} and R_t, respectively. These are the state and measurement covariance matrices. Finally, $f(\bullet)$ expresses the direct output of the second FFNN.

The proposed formulation assumes Gaussian noise characteristics, which is a standard assumption in Kalman filtering as it facilitates tractable probabilistic modeling and optimal state estimation under uncertainty [8,20,21]. Furthermore, the nonlinearity of the second FFNN, introduced by the $f(\bullet)$ function, was addressed through local linearization of the measurement equation via a first-order Taylor expansion, as reflected in the computation of the Jacobian matrix required by the EKF.

When the state-measurement model is defined, the Hybrid Extended Kalman Filter conducts the following steps:

1. Definition. Jacobian matrix of the nonlinear transition function $f(\bullet)$.

$$F_{t/t-1} = \left.\frac{\partial f_t}{\partial \mathrm{W}}\right|_{W=\hat{W}_{t/t-1}}$$

2. Initialization. For $t = 0$, set

$$\hat{W}_0 = E[W_0],$$
$$\mathrm{P}_0 = E\left[\left(W_0 - \hat{W}_0\right)\left(W_0 - \hat{W}_0\right)^T\right],$$
$$\mathrm{R}_0, Q_0$$

3. Iteration. For $t = 1, 2, \ldots$, compute the following:

a. Prediction Step:

$$\hat{W}_{t/t-1} = \hat{W}_{t-1}$$
$$P_{t/t-1} = P_{t-1} + Q_{t-1}, \tag{6}$$

b. Correction Step:

$$d_t = y_t - f_t\left(\hat{W}_{t/t-1}\right) \text{ (innovation)}$$
$$K_t = P_{t/t-1}F_{t/t-1}^T\left[F_{t/t-1}P_{t/t-1}F_{t/t-1}^T + R_t\right]^{-1} \tag{7}$$
$$\hat{W}_{t/t} = \hat{W}_{t/t-1} + K_t\left[y_t - f_t\left(\hat{W}_{t/t-1}\right)\right] \text{ (first state correction)}$$
$$\varepsilon_t = y_t - f_t\left(\hat{W}_{t/t}\right) \text{ (residual)}$$

Update the measurement covariance matrix:

$$R_t = \alpha R_{t-1} + (1-\alpha)\left(\varepsilon_t\varepsilon_t^T + F_{t/t-1}P_{t/t-1}F_{t/t-1}^T\right) \tag{8}$$
$$K_t = P_{t/t-1}F_{t/t-1}^T\left[F_{t/t-1}P_{t/t-1}F_{t/t-1}^T + R_t\right]^{-1}$$
$$P_{t/t} = \left[I - K_tF_{t/t-1}\right]P_{t/t-1} \tag{9}$$

Update the state covariance matrix:

$$Q_t = \alpha Q_{t-1} + (1-\alpha)\left(\mathrm{K}_t d_t d_t^T \mathrm{K}_t^T\right) \tag{10}$$
$$\hat{W}_{t/t} = \hat{W}_{t/t-1} + K_t\left[y_t - f_t\left(\hat{W}_{t/t-1}\right)\right] \text{ (Second State Correction)} \tag{11}$$

Equation (6) defines the error covariance matrix, which reflects the filter's confidence in its current estimates, i.e., the degree to which the estimated weights are considered optimal with respect to minimizing the training error. Equation (7) represents the Kalman gain, which constitutes a key component of the filtering process, as it governs the extent to which new observations influence the state update [27].

A critical challenge in Kalman filtering lies in the proper selection of the covariance matrices, as inappropriate choices may significantly degrade performance or even lead to filter divergence. Some approaches adopt fixed covariance matrices determined prior to the application of the algorithm [28,29], while others employ adaptive schemes that update these matrices during the filtering process, often based on recent estimates of the process and measurement noise [30,31].

However, this study employs Equations (8) and (10) to update the covariance matrices during the EKF algorithm. These formulations, originally developed by Shahrokh and Huang [32], are based on residual and innovation-based adaptive approaches, where the measurement and process noise covariance matrices are estimated through the discrepancies between observed, predicted, and filtered quantities. This adaptive nature of the covariance matrices contributes to stable convergence of the EKF training process.

Moving forward, the parameter a controls the relative contribution of past and current information, determining the degree to which the covariance matrices adapt to new data; for this reason, it is referred to as a memory parameter. A detailed analysis of these Equations can be found in Shahrokh and Huang [32] and Dona et al. [8].

As initial values for the covariance matrices, $Q_0 = I$, R_0 was initialized as the sample variance of the initial forecast errors computed over the first 24 h of the training dataset, and P_0 was initialized as a diagonal matrix with constant entries, reflecting moderate initial uncertainty in all network weights. This initialization ensures numerical stability during the early stages of the EKF iterations, while the adaptive update of the covariance matrices reduces sensitivity to the initial values.

https://doi.org/10.3390/math14101650

Once the training process of the adaptive EKF is completed, the optimal weights of the second FFNN are obtained via column vector $\hat{W}_t$ (Equation (11)). This feedforward architecture, defined by $\hat{W}_t$, utilizes the TesDs dataset to generate the improved forecasts for the wind speed at time $t+1$ based on the formula:

$$ImprovedFor_{t+1} = for_{t+1} + f_t\left(\hat{W}_t, for_t\right) \tag{12}$$

2.3. Classic and Double-Criterion Early Stop Pruning

The major drawback of the initial Hybrid Extended Kalman Filter methodology is the lack of an efficient method that mitigates the phenomenon of overfitting in its second stage. To this end, the present study modifies a pruning technique suggested by Sum et al. [33] to reduce it, thereby boosting the forecasting performance of the HEKF.

Classical pruning [34–36] is a widely used technique for reducing neural network complexity after the training process is finalized. In pruning, the contribution of each weight or an entire network connection, such as a fully connected pathway from input data through a hidden neuron to the output layer, is evaluated for its impact on model performance. Components or connections with limited contribution are permanently removed, resulting in a simplified network architecture. Through this reduction, pruning prevents the model from relying on specific patterns or memorized random noise in the training data, thereby avoiding factors that cause overfitting. Consequently, the pruned network tends to have improved generalization capabilities on unseen data.

Its effectiveness, however, depends heavily on the quality of the final model, since pruning is applied after the training process. As a result, it does not influence the learning dynamics, which could further mitigate the phenomenon of overfitting. For that reason, the present study introduces a modified pruning approach. Specifically, a double-criterion early stopping strategy was proposed, based on both the validation error and the trace of the error covariance matrix.

2.3.1. Assessing Weight Importance via Error Covariance Matrix

A key aspect of a pruning method is to quantify the contribution of each weight after the training process has been completed. To illustrate that process for the HEKF, two fundamental elements must be taken into consideration after its training: (i) the estimated state vector $\hat{W}_t = [\hat{W}_{t,1}, \ldots, \hat{W}_{t,\kappa}, \ldots, \hat{W}_{t,n_{\hat{W}_t}}]$ with $n_{\hat{W}_t}$ denoting the size of the state vector $\hat{W}_t$, and (ii) the corresponding error covariance matrix (P_t-Equation (9)). Based on this information and under the assumption of convergence [33], the asymptotic behavior of the covariance matrix can be expressed as:

$$P_\infty^{-1} = [P_\infty + Q_0]^{-1} + F_{t/t-1}F_{t/t-1}^T, \tag{13}$$

where

$$P_\infty = \lim_{t\to\infty} P_{t/t}, Q_0 = I$$

and

$$P_{t/t}^{-1} = \left[P_{t/t-1} + Q_t\right]^{-1} + F_{t/t-1}R^{-1}F_{t/t-1}^T. \tag{14}$$

Let it also be assumed that there exists a time t_1 such that, for all $t > t_1$, each P_t approaches the value of P_∞. Hence, it can be inferred that:

$$\begin{aligned} &\frac{1}{M-t_1}\sum_{t=t_1+1}^{M} \; P_\infty^{-1} - [P_\infty + Q_0]^{-1} = \\ &\frac{1}{M-t_1}\sum_{t=t_1+1}^{M} F_{t/t-1}F_{t/t-1}^T. \end{aligned} \tag{15}$$

When the size of M is sufficiently large, the term on the right side of the above equation can be approximated by its expected value. Thus, Equation (13) can be transformed to:

$$P_\infty^{-1} = [P_\infty + Q_0]^{-1} + E\Big[F_{t/t-1}F_{t/t-1}^T\Big]. \tag{16}$$

By utilizing the diagonal form of the matrix P_∞, i.e., $P_\infty = UDU^T$, Equation (16) can be rewritten as:

$$U\Big[D^{-1} - \big(D+I\big)^{-1}\Big]U^T = E\Big[F_{t/t-1}F_{t/t-1}^T\Big], \tag{17}$$

where D is a diagonal matrix consisting of the eigenvalues of P_∞:

$$D = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_{n_{\hat{W}_t}} \end{bmatrix},$$

and U the matrix which contains the corresponding eigenvectors.

Now, if β_k denotes the kth diagonal element of the matrix $\big[D^{-1} - \big(D+I\big)^{-1}\big]$, it can be written in the following form:

$$\beta_k = \frac{1}{\lambda_k} - \frac{1}{\lambda_k + 1} = \frac{1}{\lambda_k(\lambda_k+1)}. \tag{18}$$

Based on Equation (18), two special cases can be defined:

$$\beta_k \approx \begin{cases} \lambda_k^{-1}, & \text{if } \max\{\lambda_k\} \ll 1 \\ \lambda_k^{-2}, & \text{if } \min\{\lambda_k\} \gg 1 \end{cases} \tag{19}$$

Empirically, John Sum et al. [33] have shown that the eigenvalues λ_k typically exceed unity in practical applications, which justifies the approximation $\beta_k \approx 1/\lambda_k^2$. Substituting these values into Equations (16) and (17), an approximation for the quantity $E\Big[F_{t/t-1}F_{t/t-1}^T\Big]$ can be obtained:

$$\begin{aligned} E\Big[F_{t/t-1}F_{t/t-1}^T\Big] &\approx U\big[D^{-1} - (D+I)^{-1}\big]U^T \\ &= P_\infty^{-2} \end{aligned}$$

or

$$P_\infty^{-2} \approx E\Big[F_{t/t-1}F_{t/t-1}^T\Big].$$

By approximating the expected value $E\Big[F_{t/t-1}F_{t/t-1}^T\Big]$ using its mean, and substituting the definition of F_t, the following relation is obtained:

$$E\left[\left(\frac{\partial f_t(\hat{W}_t, for_t)}{\partial \hat{W}_{t,k}}\right)^2\right] \approx \left(P_\infty^{-2}\right)_{kk}, \tag{20}$$

where $(\bullet)_{kk}$. denotes the kth diagonal element.

Equation (20) quantifies how the expected change in the network output, with respect to the weight $\hat{W}_{t,k}$ of the state vector, is influenced by the asymptotic behavior of the covariance matrix. By utilizing the second-order information from the EKF algorithm, this relationship enables the definition of a weight contribution measure, which can guide the pruning process.

2.3.2. Single Weight Pruning and Extension to Connection Paths

In this work, the contribution of entire network connections was evaluated; i.e., all the weights of this path were set to zero, thereby being pruned. To illustrate that procedure, a single weight pruned was first introduced to set the basis of the algorithm. Once the training process of the hybrid filter was finalized, an approximation ($\hat{y}_t$) of the measurement equation (y_t) can be obtained. Thus, the expected predicted square error (EPSE) of this approximated function can be defined as:

$$E\left[(y_t - \hat{y}_t)^2\right] \approx E\left[\left(f_t(W_t, for_t) + v_t - f_t(\hat{W}_t, for_t)\right)^2\right] \tag{21}$$

$$\approx E\left[\left(f_t(W_t, for_t) - f_t(\hat{W}_t, for_t)\right)^2\right] + \sigma^2, \tag{22}$$

where for simplicity $\sigma^2 \equiv R_t$. It should be noted at this point that the expectation error was computed on the "future" data.

Now, consider that the kth element of state vector $\hat{W}_t$ is set to zero; i.e., $\hat{W}_{t,\kappa} = 0$, and let $\hat{y}_{t,p}$ denote the corresponding approximated function of the pruned state vector $\hat{W}_{t,p}$. The EPSE is then given by:

$$E\left[(y_t - \hat{y}_{t,p})^2\right] \approx E\left[\left(f_t(W_t, for_t) + v_t - f_t(\hat{W}_{t,p}, for_t)\right)^2\right], \tag{23}$$

or by adding the term $f_t(\hat{W}_t, for_t)$ is given by:

$$\begin{aligned} E\left[(y_t - \hat{y}_{t,p})^2\right] &\approx E\left[\left((f_t(W_t, for_t) + v_t - f_t(\hat{W}_t, for_t)) + (f_t(\hat{W}_t, for_t) - f_t(\hat{W}_{t,p}, for_t))\right)^2\right] \\ &\approx E\left[\left(f_t(W_t, for_t) - f_t(\hat{W}_t, for_t)\right)^2\right] + \sigma^2 + E\left[\left(f_t(\hat{W}_t, for_t) - f_t(\hat{W}_{t,p}, for_t)\right)^2\right]. \end{aligned} \tag{24}$$

Considering that the pruning operation introduces a small perturbation to the estimated state vector, a first-order Taylor series expansion around the unpruned solution $\hat{W}_t$ can be employed to approximate the third term on the right-hand side of Equation (24), enabling an analytical approximation of the EPSE:

$$E[(y_t - \hat{y}_{t,p})^2] \approx E\left[\left(f_t(W_t, for_t) - f_t(\hat{W}_t, for_t)\right)^2\right] + \sigma^2 + E\left[\left(\frac{\partial f_t(\hat{W}_t, for_t)}{\partial \hat{W}_{t,k}}\right)^2\right]\hat{W}_{t,k}^2$$

$$E[(y_t - \hat{y}_{t,p})^2] - E\left[\left(f_t(W_t, for_t) - f_t(\hat{W}_t, for_t)\right)^2\right] - \sigma^2 \approx E\left[\left(\frac{\partial f_t(\hat{W}_t, for_t)}{\partial \hat{W}_{t,k}}\right)^2\right]\hat{W}_{t,k}^2$$

$$\Delta E_k \approx E\left[\left(\frac{\partial f_t(\hat{W}_t, for_t)}{\partial \hat{W}_{t,k}}\right)^2\right]\hat{W}_{t,k}^2, \tag{25}$$

where ΔE_k denotes, through Equation (22), the expected error increment due to the pruned weight $\hat{W}_{t,\kappa}$. Combining Equations (20) and (25), a relation between the matrix P_∞, the vector $\hat{W}_t$, and the expected error can be constructed:

$$\Delta E_k \approx \left(P_\infty^{-2}\right)_{kk}\hat{W}_{t,k}^2 \tag{26}$$

Through Equation (26), the importance of each weight can be estimated and ranked according to their ΔE_k, producing a ranking list denoted as $\left\{\pi_1, \pi_2, \ldots, \pi_{n_{\hat{W}_t}}\right\}$, where $\Delta E_{\pi_i} \leq \Delta E_{\pi_j}$ if $i < j$.

While Equation (26) enables the assessment and ranking of individual weights, neural network structures are inherently organized in terms of interconnected pathways. Therefore, instead of treating weights independently, the proposed approach evaluated and pruned groups of weights corresponding to entire connection paths.

To present that process, let $\hat{W}_{t_a}$ denote the estimated state vector at time t_a, where the elements corresponding to the first k weights have been set to zero, while the remaining elements are identical to those in $\hat{W}_t$. The resulting incremental change in the EPSE can then be approximated as:

$$\Delta E_{[\pi_1,\pi_k]} \approx \hat{W}_{t_a}^T \left(P_\infty^{-2} \right) \hat{W}_{t_a}. \tag{27}$$

Using Equation (27), multiple weights can be pruned simultaneously, provided that $\Delta E_{[\pi_1,\pi_k]} < threshold$, thereby reducing the complexity of the neural network. It should be noted that the adopted path importance measure is based on sensitivity-related information derived from the EKF covariance matrix and does not explicitly account for the magnitude of the network weights. While this simplification ensures computational efficiency and consistency with the EKF framework, it may not fully capture the absolute contribution of each connection path. Future work will investigate hybrid importance measures combining sensitivity and weight magnitude information.

2.3.3. Double-Criterion Early Stop Pruning

Applied exclusively after the training process, classical pruning presents a notable limitation in mitigating overfitting, as it fails to exploit valuable information embedded in the learning dynamics. To address this, the present work proposes a simple yet effective modification to enhance its performance. Specifically, a double-criterion early stopping pruning strategy was introduced.

Instead of utilizing the entire training dataset (TrDs) for training the hybrid EKF, the suggested approach employed the trds and valds datasets. The former was used to update the network parameters, while the latter was employed to monitor the model performance through Equation (22). In this way, the "future data" involved in the error estimation corresponds to the validation dataset, forming the first criterion, which is defined as:

$$E_{val} = E\left[(y_t - \hat{y}_\mathrm{t})^2 \right]_{val}. \tag{28}$$

However, to ensure that the minimization of the validation error was not due to random fluctuations but reflects the actual generalization capability of the network, a second criterion was introduced, namely the minimization of the trace of the error covariance matrix:

$$Tr(P_t) = \sum_{k=1}^{n_{\hat{W}_t}} P_{kk}. \tag{29}$$

Matrix P_t expresses the confidence of the filter in its weight estimates, i.e., the extent to which the weights are minimizing the training error. Nevertheless, during the final stages of the EKF, the filter may become excessively confident about its estimates, which can lead to overfitting. The trace is adopted as a proxy for the total estimation uncertainty, as it provides a comprehensive measure of the cumulative variance across all network parameters. Unlike the determinant, which can be numerically unstable in high-dimensional weight spaces, or the spectral norm, which only accounts for the maximum eigenvalue, the trace offers a robust, computationally efficient, and holistic indicator of the filter's convergence state. This makes it particularly suitable for the proposed double-criterion early stopping, ensuring that pruning occurs only when the global uncertainty has stabilized.

Since the Hybrid Extended Kalman Filter operates in a sequential manner, a conventional early stopping strategy is not straightforward. To overcome this issue, the training process was organized into multiple passes over the trds and valds datasets, effectively emulating training epochs. At each pass, the optimal weights and the corresponding covariance matrix, based on the suggested criterion, are retained and used as the initial conditions for the subsequent pass. Consequently, the optimal set of weights was selected as:

$$\hat{W}_{t^*}, \text{where } t^* = argminE_{val} \text{ subject to } minTr(P_t),$$

ensuring both strong generalization and controlled estimation uncertainty.

The simultaneous use of two stopping criteria was designed to capture different aspects of the learning process. The validation error monitors the generalization performance, while the covariance trace reflects the convergence behavior and uncertainty evolution within the EKF framework. Therefore, they serve complementary roles, ensuring a balanced trade-off between predictive accuracy and training stability. Moreover, this controlled behavior of the proposed combined mechanism helps prevent premature convergence and mitigates potential instabilities that may arise from abrupt structural changes during pruning.

In practical implementation, the selection process was performed in two stages. The validation error was used as the primary criterion to identify the best-performing candidate models. Subsequently, among solutions with similar validation error values, the trace of the covariance matrix was employed as a secondary criterion to select the model exhibiting more stable convergence behavior.

Based on this set of weights, the pruning process was then guided by the estimated increase in prediction error, expressed through:

$$\Delta E_{[\pi_1,\pi_k]} < threshold, \tag{30}$$

where the pruning threshold is defined in a relative and adaptive manner. Specifically, in this study, a connection path i is removed if its importance measure ($\Delta E^i_{[\pi_1,\pi_\kappa]}$) is less than or equal to 20% of the mean importance ($\overline{\Delta E}$) calculated across all candidate paths in the current iteration:

$$\Delta E^i_{[\pi_1,\pi_\kappa]} \leq 0.2\,\overline{\Delta E} \tag{31}$$

This adaptive mechanism ensures that only paths with a marginal contribution to the network's predictive performance are eliminated. Particularly, the selected scaling factor provides a stringent yet flexible filter that isolates paths with negligible contribution, ensuring that the pruning process targets only redundant parameters. Moreover, by utilizing the average importance of all active connections, the proposed methodology preserves the essential non-linear mapping capabilities required for complex wind speed patterns. Finally, the value 0.2 was empirically selected to achieve a balanced trade-off between model simplification and predictive performance, while maintaining moderate pruning rates.

Overall, the proposed framework integrates covariance-based pruning with a double-criterion early stopping mechanism, enabling a dynamic trade-off between model complexity, estimation uncertainty, and generalization performance. A more analytic description of the modified Hybrid Extended Kalman Filter can be found in Appendix A.

2.4. Experimental Design

A time-window application is used to assess the performance of the proposed methodology. Specifically, the necessary forecasts for the weather parameter of 10 m wind speed are produced by the operational WRF model [37–39], particularly from its second nested domain (d02), with a horizontal grid spacing of 5 km, while the corresponding recorded observations are obtained from the meteorological station of the Department of Meteorology and Climatology (https://meteo.geo.auth.gr/en/; last accessed on 18 April 2026) in the School of Geology of the Aristotle University of Thessaloniki (AUTH; 40.63177° N, 22.95755° E). The analysis focuses on 10 m wind speed, as it constitutes a highly variable atmospheric parameter that is challenging to predict due to its sensitivity to both synoptic and local-scale processes. Furthermore, the AUTH station was employed, as it ensures a high-quality dataset without missing values.

The designed process produces improved forecasts based on a sequential pattern. Specifically, in each time window (TW), the available data were partitioned into the TrDs and the TesDs datasets. The TrDs dataset was employed to train the hybrid filter, whereas the TesDs was applied to generate the improved forecasts. Once the recorded observations corresponding to the forecasting period defined by the TesDs become fully available, they are incorporated into the TrDs, and the process advances to the next time window, where the HEKF is retrained using the updated TrDs, producing a new forecasting interval.

This sequential retraining approach effectively addresses the non-stationarity of wind speed by allowing the hybrid model to dynamically adapt to evolving atmospheric regimes and local-scale processes. This approach ensures that the hybrid scheme avoids the pitfall of overfitting to outdated patterns, maintaining high generalization even with relatively short training datasets.

In this work, each time window utilized 1 week of data for training and 1 day for testing. Each day consisted of 19 hourly records, as the forecasting procedure starts from the 6th forecast hour of the WRF model (due to the WRF spin-up time). Thus, each TW was trained using 133 h of data (TrDs: 133 h, split ~80/20 into trds/valds) and produces improved predictions for the next 19 h (TesDs: 19 h, a 1-day forecast horizon). This process was repeated for six consecutive time windows, corresponding to a total of six days. The algorithmic process of the time-window configuration is summarized in Appendix B.

To quantify the improvement achieved by the different strategies, three statistical metrics were utilized: BIAS, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). The joint use of these metrics ensures a thorough evaluation of forecasting performance. Particularly, BIAS quantifies systematic deviations, with positive (negative) values corresponding to underestimation (overestimation) by the model, whereas MAE and RMSE provide complementary information on the non-systematic component of the forecast error.

MAE reflects the average magnitude of the random deviations, treating all errors equally, while RMSE emphasizes larger deviations, highlighting the impact of occasional large errors on overall predictive performance. This combined assessment facilitates a deeper understanding of the extent to which the proposed filtering approach improves both the accuracy and robustness of the WRF model.

3. Results and Discussion

This section presents the results obtained from the various methodologies in terms of computational efficiency and predictive accuracy. Table 2 summarizes the average statistical metrics across the six time windows of the first five months of 2020, while Figure 5 illustrates the computational efficiency, expressed in terms of the computational requirements of the Classic and Early Stop Pruning methods.

Table 2. Evaluation indicators of the 10m wind speed forecasts (m/s). Average results.

	WRF	Initial EKF	Classic Pruning	Early Stop Pruning
BIAS				
January	−2.7400	−0.8929	−0.6900	0.2759
February	−4.2356	0.1160	0.4146	0.3356
March	−4.1367	−0.9246	0.5504	0.0497
April	−2.3448	−0.3307	0.2496	−0.0481
May	−1.7587	−0.4292	−0.7148	−0.1741
MAE				
January	2.7795	2.3145	1.6620	0.7512
February	4.2356	1.3950	1.0034	0.9242
March	4.1446	2.8500	1.2285	0.8341
April	2.3696	1.9080	0.9709	0.8362
May	1.7722	0.9068	1.4104	0.5764
RMSE				
January	3.1089	2.6007	1.8209	0.9386
February	4.5890	1.7533	1.1733	1.1201

Table 2. *Cont.*

	WRF	Initial EKF	Classic Pruning	Early Stop Pruning
March	4.7276	3.1977	1.3722	1.0543
April	2.6436	2.1753	1.1901	1.0329
May	1.9953	1.0607	1.5192	0.6994

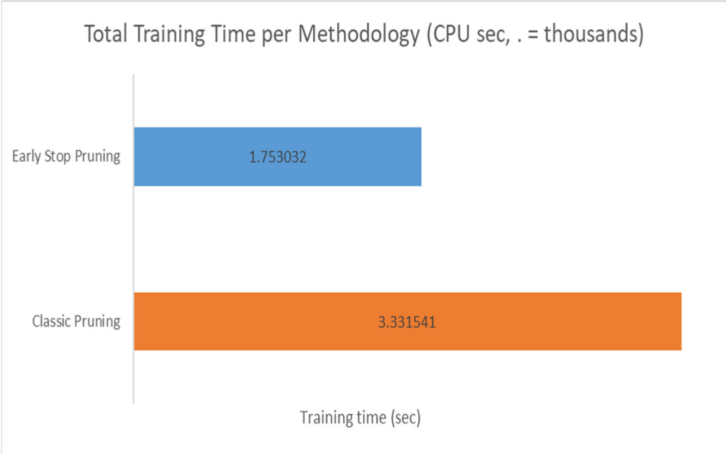

Figure 5. Total training time for January–May. Early Stop Pruning vs. Classic Pruning.

Figures 6–8 depict the time series forecasts, and Figures 9–11 present the corresponding boxplots of the forecasting errors (observation-forecast) for January, March, and May. These diagrams provide a visual comparison of forecast accuracy across the different methods and study periods, along with the corresponding quantitative assessment of error distributions. Specifically, the time series describes to what extent the produced forecasts capture the distribution of the weather parameter under study, while the boxplots indicate the forecast quality by measuring the variability of the forecast error.

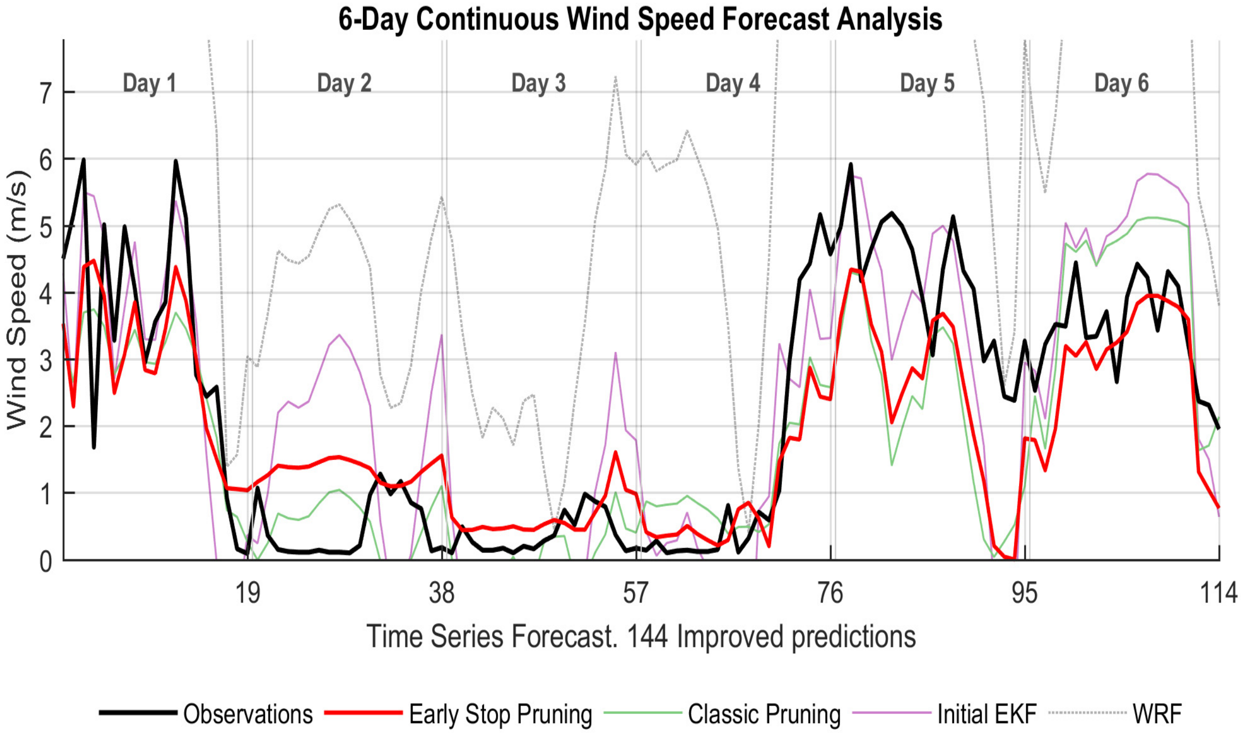

Figure 6. Time series forecast. February 2020. AUTH station.

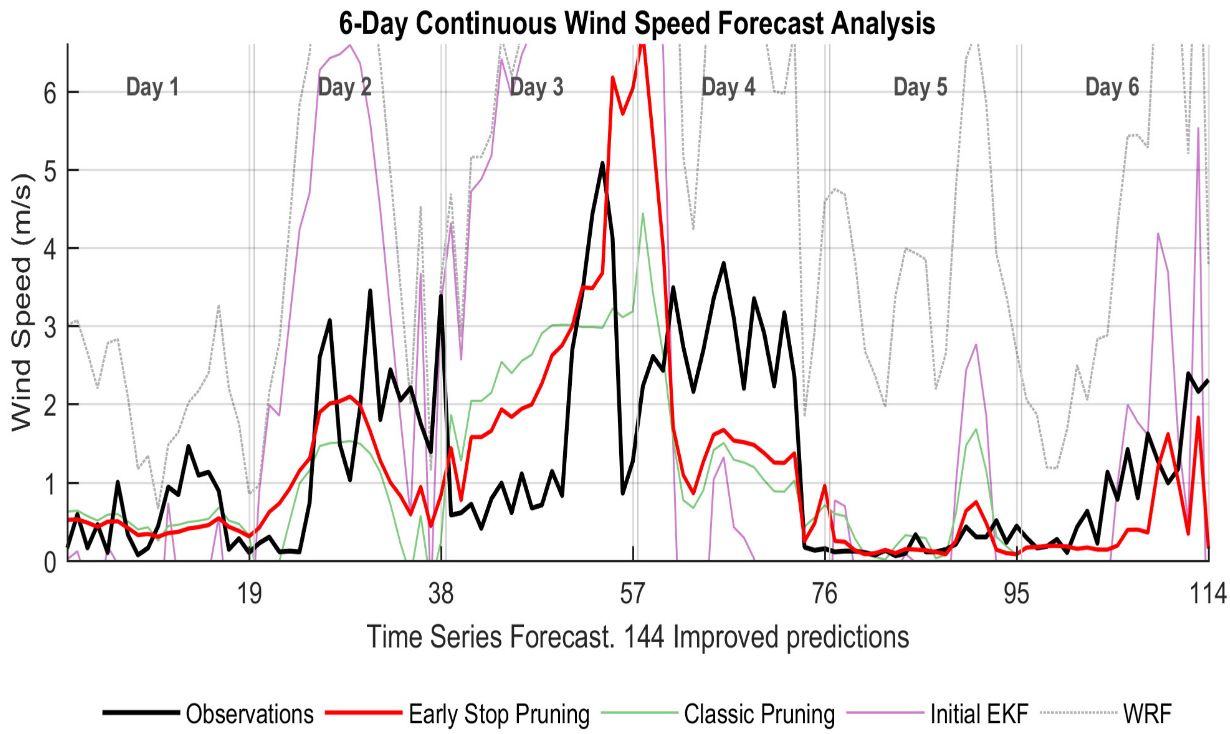

Figure 7. Time series forecast. March 2020. AUTH station.

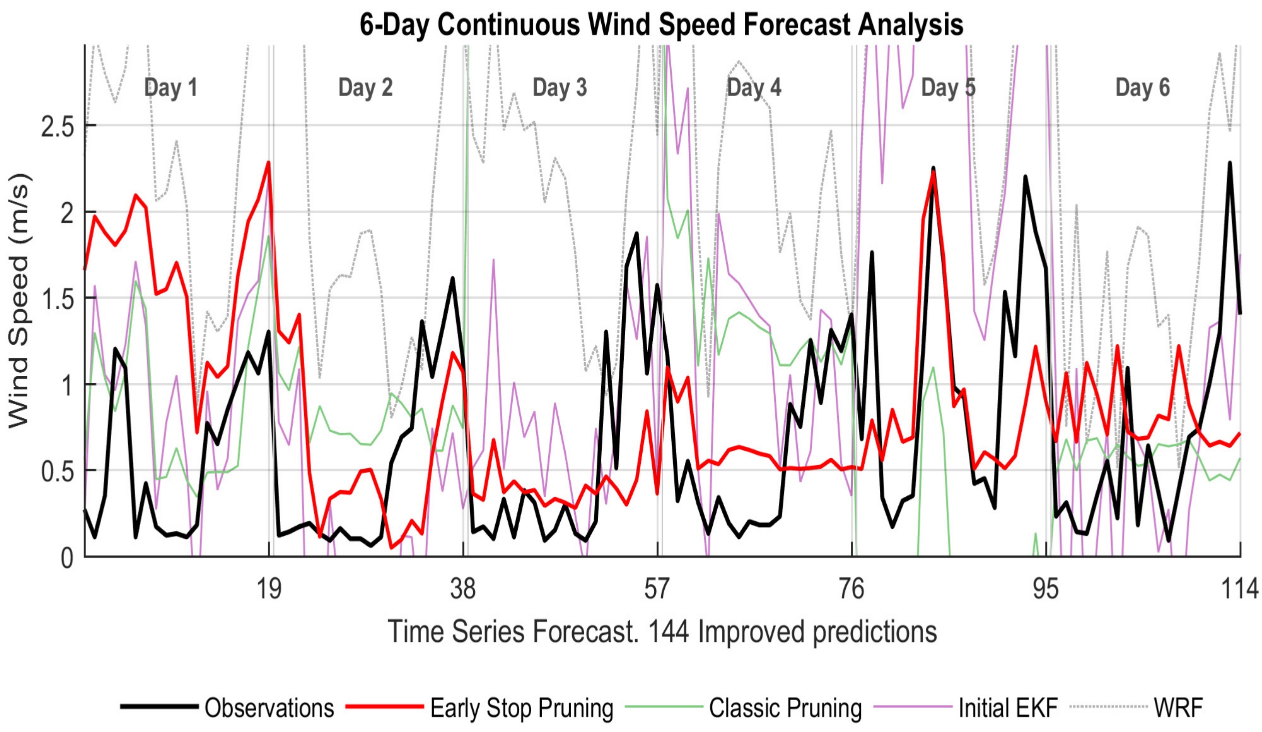

Figure 8. Time series forecast. May 2020. AUTH station.

Starting from the first week of each month, the time-window process is applied sequentially, producing consecutive daily forecasts, resulting in six forecasted days. Therefore, the horizontal axis in Figures 6–8 presents these forecasted days, rather than the entire monthly period.

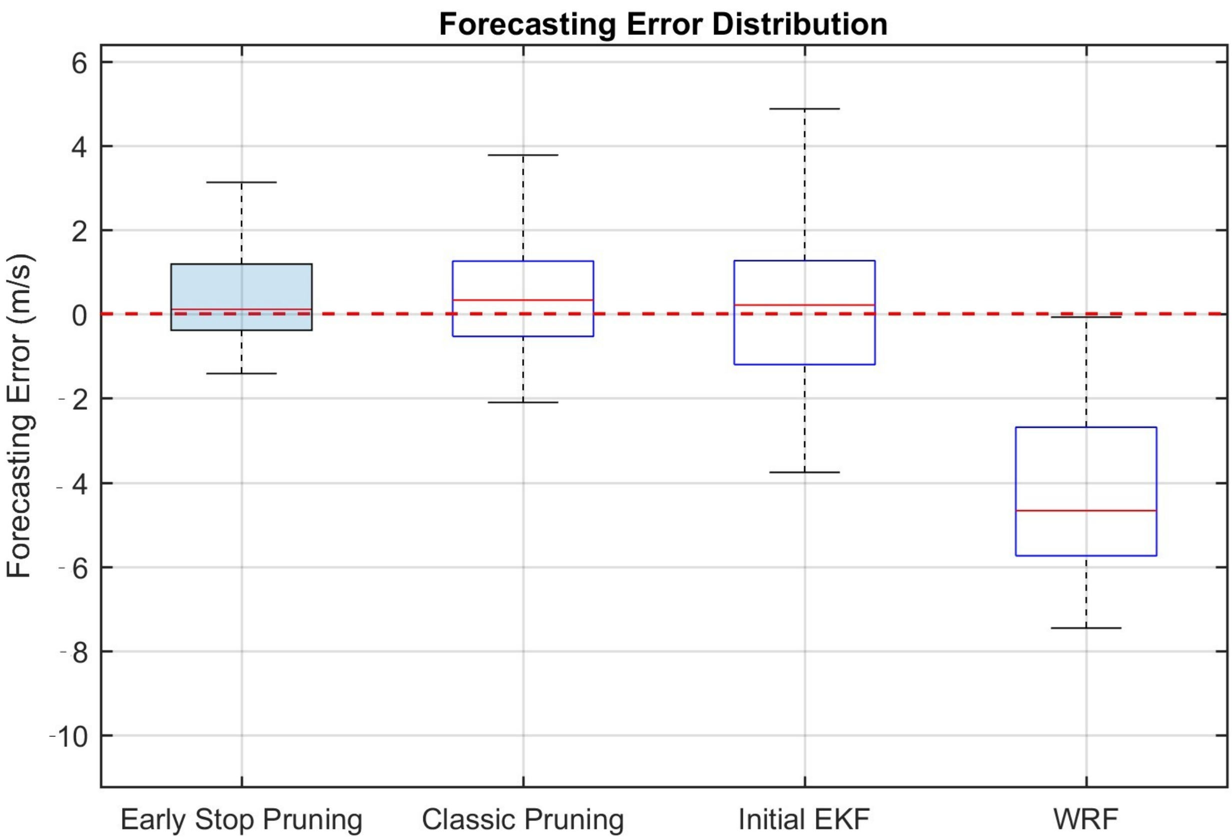

Figure 9. Boxplot of forecasting errors. February 2020. AUTH station.

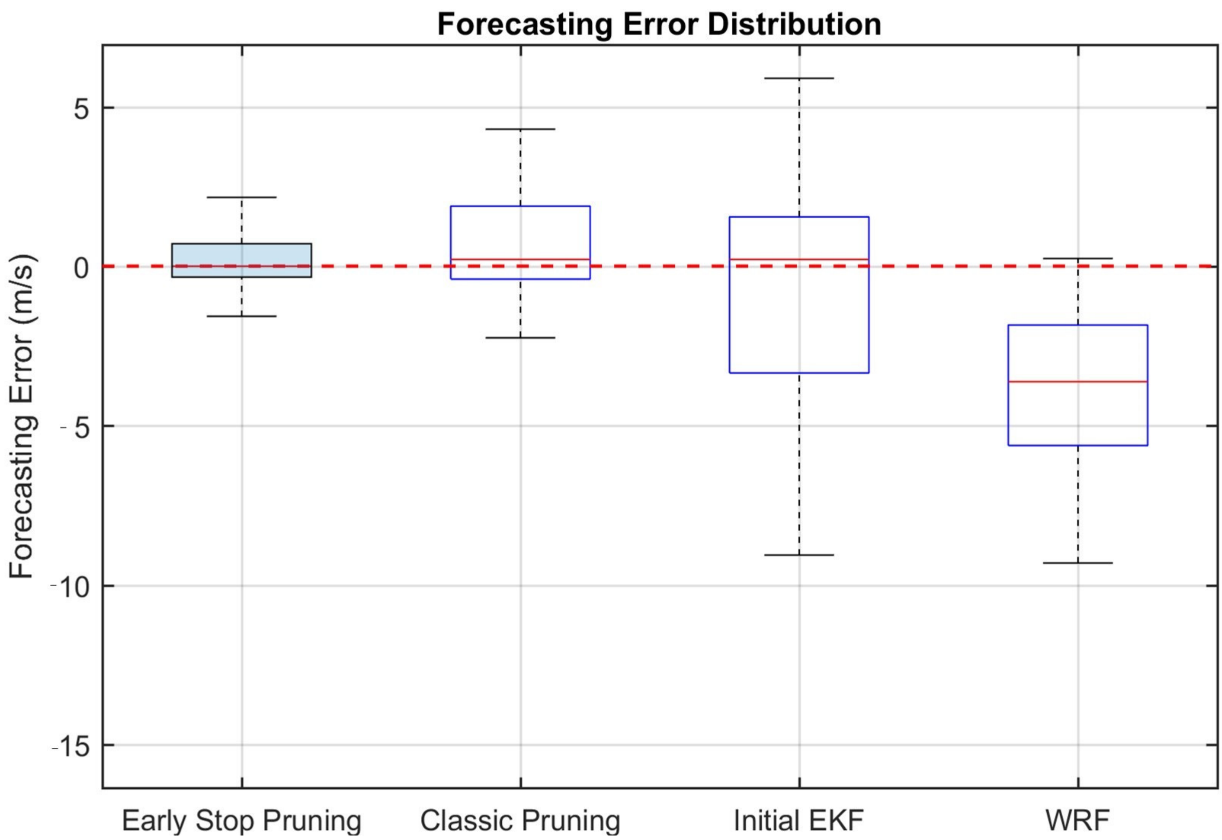

Figure 10. Boxplot of forecasting errors. March 2020. AUTH station.

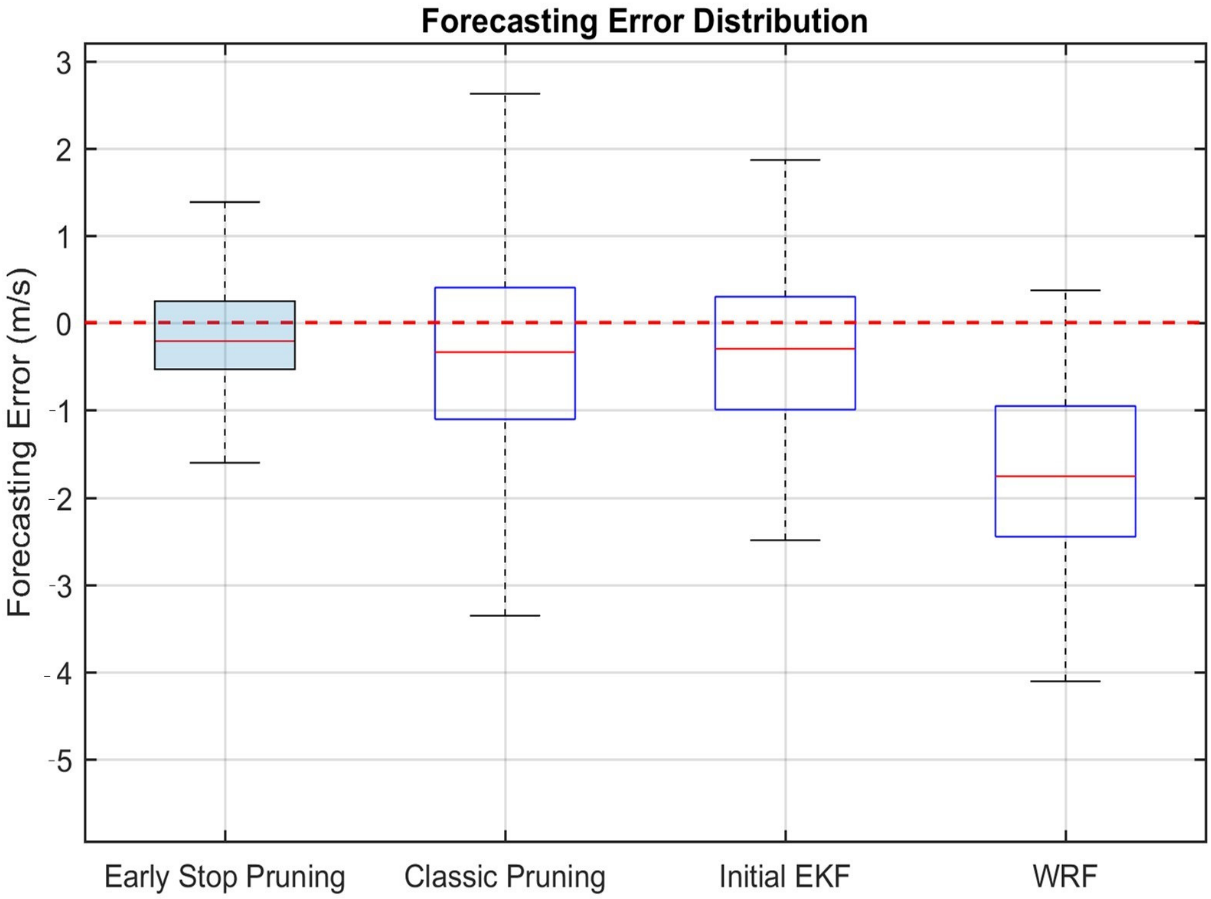

Figure 11. Boxplot of forecasting errors. May 2020. AUTH station.

The overall evaluation period spans the first five months of 2020 (January–May), covering a wide range of atmospheric conditions and significant variability in wind patterns across 570 forecast instances (5 months $\times$ 6 days $\times$ 19 h). However, January, March, and May are selected as representative cases, as they effectively capture the diversity of the prevailing meteorological regimes (winter: Jan synoptic dominance; spring: Mar/May thermal/local influences). The terms Observations, Early Stop Pruning, Classic Pruning, and Initial EKF refer to the recorded data, the proposed approach, the classical pruning technique, and the initial hybrid EKF scheme, respectively. Finally, all examined approaches are trained under identical conditions, including the same number of epochs, identical pruning threshold and scaling factor, and the same training and validation datasets.

The comparative assessment of the statistical metrics, as presented in Table 2, reveals the enhanced predictive capabilities of the Early Stop Pruning method across all timeframes under consideration. A key observation is the substantial decrease in systematic error (BIAS). While the unrefined WRF model demonstrates a considerable and persistent overestimation of wind speed—exemplified by values as high as -4.24 m/s in February—the proposed methodology effectively reduces the BIAS to 0.3356 m/s in February and adjusts it to a value remarkably close to zero in other months (e.g., 0.0497 m/s in March and -0.0481 m/s in April). Consequently, this indicates that the suggested approach successfully mitigates the intrinsic "offset" of the physical model, thereby yielding nearly unbiased forecasts.

Furthermore, the evaluation of non-systematic errors through the MAE and RMSE metrics confirms the robustness of the proposed strategy. The Early Stop Pruning achieves the lowest MAE in the vast majority of cases, reaching a minimum of 0.58 m/s in May, which highlights its high precision in hourly intervals. More importantly, the notable decrease in RMSE compared to both the Initial EKF and the Classic Pruning (e.g., 0.94 m/s vs. 1.82 m/s in January) suggests that the proposed model is less prone to large forecasting outliers.

Quantitatively, the proposed approach achieves significant reductions, with RMSE reductions ranging from approximately 60% to 78% and MAE reductions between 65% and 80%, depending on the examined period, when compared to the initial WRF model output. This effectively bridges the gap between coarse physical simulations and local-scale observations. These significant enhancements can be partly attributed to the inherent limitations of the WRF model, which introduce systematic and non-systematic errors that affect the quality of its forecasts. Such errors arise from approximations in physical parameterizations, boundary conditions, and model resolution, leading to biases and reduced accuracy, particularly under complex atmospheric conditions.

Compared to the Classic Pruning technique, the Early Stop strategy generally leads to lower errors, although minor deviations may occur in specific cases. These findings, in conjunction with the computational efficiency analyzed in Figure 5, indicate that the double-criterion early stopping pruning not only mitigates overfitting, ensuring high-precision and stable final forecasts, but also leads to reduced computational requirements, making it suitable for real-time applications.

The visual examination of the time series diagrams (Figures 6–8) confirms the superiority of the Early Stop Pruning method compared to other techniques. The proposed approach (red line) accurately reflects the observed data (black line), successfully avoiding the major overestimations seen in the raw WRF model (gray line) and the instabilities in the Initial EKF (purple line). In particular, during times of rapid wind speed changes—like Day 4 in February and Day 3 in March—the Early Stop method demonstrates a better ability to adapt to sudden shifts while maintaining a smooth and realistic forecasting trend. In contrast, the Classic Pruning (green line) often displays either delayed responses or unnatural swings during these transition periods.

A key benefit of the Early Stop Pruning strategy is its ability to effectively capture wind speed maxima events. In the 6-day continuous diagrams, the proposed model accurately represents peak wind speed values, usually with smaller "overshooting" or "undershooting" than the ones found in the Initial EKF and the raw WRF data. While the WRF model often makes unrealistic predictions, the Early Stop method connects the forecast with the physical limits of the local environment. During low wind speed periods, the proposed system shows strong stability, while the Classic Pruning often experiences erratic changes when wind speeds are near zero. The double-criterion approach ensures that the neural network maintains adequate structural complexity to accurately model low-wind conditions while removing unnecessary neurons that could add noise.

The analysis of error distribution (Figures 9–11) supports these conclusions. Early Stop Pruning shows the most concentrated error distribution, with the Interquartile Range (IQR) being the narrowest among all the models compared. Furthermore, the median value closely matches the zero-error dashed line. This visually confirms that the proposed method effectively removes systematic bias and ensures high forecast consistency by significantly reducing the variance of the residuals. Additionally, the decrease in forecasting outliers is clear in the boxplot diagrams, where the "whiskers" of the proposed method are much shorter. In the March dataset, Early Stop Pruning keeps the error range within an exceptionally tight envelope. This stability, along with the previously analyzed computational efficiency, establishes the proposed method as a reliable option for real-time applications, ensuring accuracy across the entire operational range without high computational costs.

Summarizing the comparative analysis, the proposed double-criterion Early Stop Pruning technique demonstrates substantial improvements in meteorological time series forecasting, maintaining a computationally efficient framework. It yields consistent enhancements, as shown in Table 2, across all statistical performance metrics, effectively capturing the wind speed distribution, including periods of rapid temporal variability and abrupt transitions, thereby providing an accurate representation of maximum wind speed events. These findings indicate the consistent superiority, particularly in terms of significant bias reduction and improved MAE and RMSE values, highlighting its efficiency under varying conditions. Additionally, it should be highlighted that the observed improvements are consistent across all examined time periods, indicating that the performance gains are systematic rather than due to random fluctuations, with no notable variability observed during repeated executions of the training process.

Furthermore, the proposed methodology preserves a significantly greater segment of the network, constraining the reduction to a mere 2% to 13%, maintaining the critical non-linear mapping functionalities essential for modeling fluctuating wind patterns. This data-driven "intelligence" mitigates the excessive pruning seen in the conventional model, guaranteeing the network's continued resilience in capturing the intricate dynamics revealed through time series analysis.

Importantly, the proposed hybrid scheme addresses a critical limitation of classical Kalman Filter-based approaches. Although traditional Kalman filtering is highly effective in reducing systematic bias, it often exhibits limited capability in reducing the variability of the forecast error, as measured by the RMSE and MAE, particularly in highly stochastic environments such as wind speed forecasting [40,41]. The suggested strategy, on the contrary, goes beyond that simple bias correction, improving both generalization and stability through the combined use of statistical and machine learning tools, leading to a more robust, balanced, and reliable forecasting performance compared to conventional methodologies.

4. Conclusions

In this study, a novel hybrid methodology was developed to enhance the accuracy of wind speed forecasting by integrating a Hybrid Extended Kalman Filter with an "Early Stop Pruning" mechanism. The proposed approach addresses the inherent limitations of Numerical Weather Prediction models, such as the WRF, which often exhibit increased forecast errors due to local topographic and climatic complexities.

The core contribution of this work lies in the development of a sequential pruning strategy that operates dynamically during the learning phase. By employing a dual-criterion evaluation based on validation error and the trace of the error covariance matrix, the model achieves a refined neural architecture that minimizes overfitting while maintaining high computational efficiency.

The obtained results demonstrate a significant improvement in forecasting performance. Specifically, the proposed approach reduces the RMSE and MAE up to 78% and 80%, respectively, compared to the raw WRF output. Furthermore, the BIAS was effectively neutralized, bringing the systematic error remarkably close to zero, indicating that the proposed hybrid scheme effectively constrains both types of forecast error. From a computational perspective, the "Early Stop" mechanism reduced the training time by nearly 50% compared to standard pruning methods, confirming its suitability for real-time operational applications. Overall, the proposed methodology provides a robust and efficient solution for wind energy forecasting.

Future research will focus on comparing the suggested hybrid scheme with modern deep learning architectures, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), to further assess its performance and generalization capabilities.

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

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Abbreviations and Nomenclature

The following abbreviations and nomenclature are used in this manuscript:

Abbreviations	
ANNs	Artificial Neural Networks
EKF	Extended Kalman Filter
EPSE	Expected Predicted Square Error
FFNNs	FeedForward Neural Networks
HEKF	Hybrid Extended Kalman Filter
LM	Levenberg–Marquardt
MAE	Mean Absolute Error
MSE	Mean Squared Error
pFFNN	parametrized FeedForward Neural Network
RMSE	Root Mean Square Error
TesDs	Testing Dataset for Hybrid Filter Performance
TrDs	Training Dataset for the Hybrid Filter
trds	Training Dataset for the pFFNN
valds	validation Dataset for the pFFNN
WRF	Weather Research and Forecasting
Nomenclature	
t	Time
φ	Activation function for the FFNNs
X	Input matrix for the pFFNN
X_t	Input vector
N	Size of the TrDs
for_t	WRF forecast
a	Memory parameter
er^t	Training error
T_t	Target
W	Column (State) vector of FFNN variables
J	Jacobian matrix
er	Vectorize training error
I	Identity matrix
μ	Combination parameter
$d_{s,l}$	Synaptic weight between the s input and the l neuron in the Hidden layer
b_l	Bias of the l neuron
v_l	Synaptic weight between l neuron and the output layer

net_t^l	Input of the l neuron based on the t training pattern
b_{out}	Bias of the output layer
y_t	Forecast error
$\hat{y}_t$	Expected forecast error
f	Direct output of the second FFNN
w	State noise
v	Measurement noise
Q	Process noise covariance matrix
R	Measurement noise covariance matrix
P	Error covariance matrix
F	Jacobian matrix of the transition function f
d	Innovation
ε	Residual
K	Kalman gain
$ImprovedFor$	Improved forecasts
$\hat{W}$	State vector after the HEKF implementation
$n_{\hat{W}_t}$	Size of the state vector
P_∞^{-1}	Asymptotic behavior of the covariance matrix
$\hat{W}_{t,\kappa}$	k Pruned weight
ΔE_k	Expected error increment of the pruned weight k
$\left\{\pi_1, \pi_2, \ldots, \pi_{n_{\hat{W}_t}}\right\}$	Ranking list
E_{val}	Validation error
$Tr(P_t)$	Trace of the error covariance matrix
$\hat{W}_{t^*}$	Optimum weights from the double-criterion pruning

Appendix A. Stages of the Hybrid Extended Kalman Filter

Algorithm A1: Stage 1. pFFNN

```
Based on TrDs create trds and valds:
    for each parameter a do
        Insert parameter into the trds
         for each Neuron do
            Create the network.
                while train ≤ maxtrain % For every network, conduct
                multiple trainings
                    performance → Network's performance % MSE
                    base on valds.
                    if performance < Initial Value
                        Initial Value →
                        performance
                    endif
                    train → train + 1
                endwhile
                train → 1
        endfor
        readjust Initial Value
    endfor
```

Define the optimal parameter and number of neurons

```
position → the index with minimum MSE
Best Neurons → Neuron (position) and Best Parameter → parameter (position)
```

Algorithm A2: Stage 2. Adaptive EKF with Double-Criterion pruning-Second FFNN	
Based on the memory parameter a and the number of hidden neurons:	Algorithm A1
Utilize trds and valds datasets	Same as Algorithm A1
Create the initial values for the hybrid Filter:	W_0, P_0, Q_0, R_0
while $\leq$ *nepochs* % Until the specified epochs are reached	
for each training data in trds do	
Implement Hybrid Extended Kalman Filter with double-criterion early stop and valds	
if double-criterion is satisfied	
Store $\hat{W}_{t^*}$, P_{t^*}, Q_{t^*} and R_{t^*}	
endif	
endfor	
set $W_0 = \hat{W}_{t^*}$, $P_0 = P_{t^*}$, $Q_0 = Q_{t^*}$, $R_0 = R_{t^*}$	
endwhile	
Obtain the optimal state vector $\hat{W}_{t^*}$	

Appendix B. Structure of the Time-Window Process

Algorithm A3: Time-Window Process	
Load the data:	{Inputs, Targets} $\rightarrow$ {WRF forecasts, AUTH observations}
Define the TrDs and the TesDs datasets	
Define the maximum number of time windows:	$Max_{tw} = 6$
Define the set of parameters:	$a \rightarrow [0.3 : 0.01 : 0.4]$
Define the set of Neurons	$[20, \ldots, 30]$
Define the maximum number of epochs	$nepochs = 10$
for $k = 1, \ldots, Max\_tw$	
Apply Algorithm A1 using TrDs dataset	Obtain the optimal combination of neurons and value of the memory parameter.
Apply Algorithm A2 using TrDs dataset.	Obtain the optimal vector $\hat{W}_{t^*}$.
Generate the improved predictions using TesDs	
Evaluate the method based on TesDs	Bias, MAE and RMSE indices.
Update the TrDS when next day's observations are available	
end	
Store the aggregate results from every time window	

References

1. Kariniotakis, G.N.; Pinson, P. Evaluation of the MORE-CARE Wind Power Prediction Platform. Performance of the Fuzzy Logic Based Models. In Proceedings of the EWEC 2003—European Wind Energy Conference, Madrid, Spain, 16–19 June 2003.
2. Kariniotakis, G.; Martí, I.; Casas, D.; Pinson, P.; Nielsen, T.S.; Madsen, H.; Giebel, G.; Usaola, J.; Sanchez, I. What Performance Can Be Expected by Short-Term Wind Power Prediction Models Depending on Site Characteristics? In Proceedings of the EWC 2004 Conference, Tokyo, Japan, 2–4 August 2004; pp. 22–25.

3. Vanem, E. Long-Term Time-Dependent Stochastic Modelling of Extreme Waves. *Stoch. Environ. Res. Risk Assess.* **2011**, *25*, 185–209. [CrossRef]
4. Giebel, G. *On the Benefits of Distributed Generation of Wind Energy in Europe*; VDI-Verlag: Dusseldorf, Germany, 2001.
5. Resconi, G. Geometry of Risk Analysis (Morphogenetic System). *Stoch. Environ. Res. Risk Assess.* **2009**, *23*, 425–432. [CrossRef]
6. Chang, Y.; Yang, H.; Chen, Y.; Zhou, M.; Yang, H.; Wang, Y.; Zhang, Y. A Hybrid Model for Long-Term Wind Power Forecasting Utilizing NWP Subsequence Correction and Multi-Scale Deep Learning Regression Methods. *IEEE Trans. Sustain. Energy* **2024**, *15*, 263–275. [CrossRef]
7. Aminjan, K.K.; Sedaghat, M.; Heidari, M.; Khashehchi, M.; Mohammadzadeh, K.; Salahinezhad, M.; Bina, R. Numerical investigation of the impact of fuel temperature on spray characteristics in a pressure-swirl atomizer with spiral path. *Exp. Comput. Multiph. Flow* **2024**, *6*, 428–445. [CrossRef]
8. Donas, A.; Galanis, G.; Pytharoulis, I.; Famelis, I.T. A Hybrid Extended Kalman Filter Based on Parametrized ANNs for the Improvement of the Forecasts of Numerical Weather and Wave Prediction Models. *Atmosphere* **2024**, *15*, 828. [CrossRef]
9. Jabbar, H.K.; Khan, R.Z. Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study). In *Computer Science, Communication and Instrumentation Devices*; Research Publishing Services: Singapore, 2014; pp. 163–172. [CrossRef]
10. Berner, J.; Grohs, P.; Kutyniok, G.; Petersen, P. The Modern Mathematics of Deep Learning. In *Mathematical Aspects of Deep Learning*, 1st ed.; Grohs, P., Kutyniok, G., Eds.; Cambridge University Press: Cambridge, UK, 2022; pp. 1–111. [CrossRef]
11. Rajnikant, P.N.; Khanna, R. Mathematical Models of Neural Networks: Bridging the Gap between Biology and Computation. *Int. J. Sci. Res. Sci. Technol.* **2025**, *12*, 1156–1161. [CrossRef]
12. Madhiarasan, M.; Louzazni, M. Analysis of Artificial Neural Network: Architecture, Types, and Forecasting Applications. *J. Electr. Comput. Eng.* **2022**, *2022*, 5416722. [CrossRef]
13. Anderson, J.A. *An Introduction to Neural Networks*; The MIT Press: Cambridge, MA, USA, 1995. [CrossRef]
14. Stăvărache, G.; Ciortan, S.; Rusu, E. Optimization of Artificial Neural Networks Based Models for Wave Height Prediction. *E3S Web Conf.* **2020**, *173*, 03007. [CrossRef]
15. Bui, N.T.; Hasegawa, H. Training Artificial Neural Network Using Modification of Differential Evolution Algorithm. *Int. J. Mach. Learn. Comput.* **2015**, *5*, 1–6. [CrossRef]
16. Rauf, H.T.; Bangyal, W.H.; Ahmad, J.; Bangyal, S.A. Training of Artificial Neural Network Using PSO With Novel Initialization Technique. In *Proceedings of the 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhier, Bahrain, 20–21 December 2020*; IEEE: Sakhier, Bahrain, 2018; pp. 1–8. [CrossRef]
17. Hagan, M.T.; Demuth, H.B.; Beale, M.H.; De Jesús, O. *Neural Network Design*, 2nd ed.; Martin Hagan: Stillwater, OK, USA, 2014.
18. Yu, H.; Wilamowski, B. Levenberg–Marquardt Training. In *Intelligent Systems*; Irwin, J., Ed.; Electrical Engineering Handbook; CRC Press: Boca Raton, FL, USA, 2011; pp. 1–16. [CrossRef]
19. Ying, X. An Overview of Overfitting and Its Solutions. *J. Phys. Conf. Ser.* **2019**, *1168*, 022022. [CrossRef]
20. Ribeiro, M.I. Kalman and Extended Kalman Filters: Concept, Derivation and Properties. *Inst. Syst. Robot.* **2004**, *43*, 3736–3741.
21. Hur, S. Short-Term Wind Speed Prediction Using Extended Kalman Filter and Machine Learning. *Energy Rep.* **2021**, *7*, 1046–1054. [CrossRef]
22. Chernodub, A. Direct Method for Training Feed-forward Neural Networks using Batch Extended Kalman Filter for Multi-Step-Ahead Predictions. *arXiv* **2016**, arXiv:1605.03764. [CrossRef]
23. Ciocoiu, I.B. RBF Networks Training Using a Dual Extended Kalman Filter. *Neurocomputing* **2002**, *48*, 609–622. [CrossRef]
24. De Lima, D.P. Neural Network Training Using Unscented and Extended Kalman Filter. *Robot. Autom. Eng. J.* **2017**, *1*. [CrossRef]
25. Wang, J.; Zhu, L.; Cai, Z.; Gong, W.; Lu, X. Training RBF Networks with an Extended Kalman Filter Optimized Using Fuzzy Logic. In *Intelligent Information Processing III*; Shi, Z., Shimohara, K., Feng, D., Eds.; Springer: Boston, MA, USA, 2007; pp. 317–326. [CrossRef]
26. Puskorius, G.V.; Feldkamp, L.A. Extensions and Enhancements of Decoupled Extended Kalman Filter Training. In *Proceedings of International Conference on Neural Networks (ICNN'97), Houston, TX, USA, 9–12 June 1997*; IEEE: Houston, TX, USA, 1997; Volume 3, pp. 1879–1883. [CrossRef]
27. Galanis, G.; Anadranistakis, M. A One-dimensional Kalman Filter for the Correction of near Surface Temperature Forecasts. *Meteorol. Appl.* **2002**, *9*, 437–441. [CrossRef]
28. Homleid, M. Diurnal Corrections of Short-Term Surface Temperature Forecasts Using the Kalman Filter. *Weather Forecast.* **1995**, *10*, 689–707. [CrossRef]
29. Libonati, R.; Trigo, I.; DaCamara, C.C. Correction of 2 M-Temperature Forecasts Using Kalman Filtering Technique. *Atmos. Res.* **2008**, *87*, 183–197. [CrossRef]
30. Galanis, G.; Chu, P.C.; Kallos, G. Statistical Post Processes for the Improvement of the Results of Numerical Wave Prediction Models. A Combination of Kolmogorov-Zurbenko and Kalman Filters. *J. Oper. Oceanogr.* **2011**, *4*, 23–31. [CrossRef]

31. Xu, J.; Xiao, Z.; Lin, Z.; Li, M. System Bias Correction of Short-Term Hub-Height Wind Forecasts Using the Kalman Filter. *Prot. Control Mod. Power Syst.* **2021**, *6*, 37. [CrossRef]
32. Akhlaghi, S.; Zhou, N.; Huang, Z. Adaptive Adjustment of Noise Covariance in Kalman Filter for Dynamic State Estimation. In *Proceedings of the 2017 IEEE Power & Energy Society General Meeting, Chicago, IL, USA, 16–20 July 2017*; IEEE: Chicago, IL, USA, 2017; pp. 1–5. [CrossRef]
33. Sum, J.; Leung, C.-S.; Young, G.H.; Kan, W.-K. On the Kalman filtering method in neural network training and pruning. *IEEE Trans. Neural Netw.* **1999**, *10*, 161–166. [CrossRef]
34. Finnoff, W.; Hergert, F.; Zimmermann, H.G. Improving model selection by nonconvergent methods. *Neural Netw.* **1993**, *6*, 771–783. [CrossRef]
35. Hassibi, B.; Stork, D.G. Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. *Adv. Neural Inform. Proc. Syst.* **1992**, *5*, 164–171.
36. Lecun, Y.; Denker, J.; Solla, S. Optimal Brain Damage. *Adv. Neural Inf. Process. Syst.* **1989**, *2*, 605.
37. Skamarock, W.C.; Klemp, J.B.; Dudhia, J.; Gill, D.O.; Liu, Z.; Berner, J.; Wang, W.; Powers, J.G.; Duda, M.G.; Barker, D.M.; et al. *A Description of the Advanced Research WRF Model Version 4*; UCAR/NCAR: Boulder, CO, USA, 2019. [CrossRef]
38. Androulidakis, Y.; Makris, C.; Kolovoyiannis, V.; Kombiadou, K.; Krestenitis, Y.; Kartsios, S.; Pytharoulis, I.; Baltikas, V.; Mallios, Z. Operational platform for metocean forecasts in Thermaikos Gulf (Aegean Sea, Greece). *J. Oper. Oceanogr.* **2025**, *18*, 123–149. [CrossRef]
39. Androulidakis, Y.; Makris, C.; Mallios, Z.; Pytharoulis, I.; Baltikas, V.; Krestenitis, Y. Storm surges and coastal inundation during extreme events in the Mediterranean Sea: The IANOS Medicane. *Nat. Hazards* **2023**, *117*, 939–978. [CrossRef]
40. Cassola, F.; Burlando, M. Wind speed and wind energy forecast through Kalman filtering of Numerical Weather Prediction model output. *Appl. Energy* **2012**, *99*, 154–166. [CrossRef]
41. Pulido, M.; Tandeo, P.; Bocquet, M.; Carrassi, A.; Lucini, M. Stochastic parameterization identification using ensemble Kalman filtering combined with expectation-maximization and Newton-Raphson maximum likelihood methods. *arXiv* **2017**, arXiv:1709.07328. [CrossRef]