frontiers | Frontiers in Energy Research

TYPE Brief Research Report
PUBLISHED 12 April 2023
DOI 10.3389/fenrg.2022.1047381

Check for updates

OPEN ACCESS

EDITED BY
Xun Shen,
Tokyo Institute of Technology, Japan

REVIEWED BY
Yan Zhang,
Tokyo University of Agriculture and Technology, Japan
Shuang Zhao,
Hefei University of Technology, China

*CORRESPONDENCE
Hairong Wang,
36441472@qq.com

SPECIALTY SECTION
This article was submitted to Smart Grids, a section of the journal Frontiers in Energy Research

RECEIVED 18 September 2022
ACCEPTED 21 October 2022
PUBLISHED 12 April 2023

CITATION
Wang H (2023), Extreme learning Kalman filter for short-term wind speed prediction.
Front. Energy Res. 10:1047381.
doi: 10.3389/fenrg.2022.1047381

Extreme learning Kalman filter for short-term wind speed prediction

Hairong Wang*

College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Accurate prediction of wind speed is critical for realizing optimal operation of a wind farm in real-time. Prediction is challenging due to a high level of uncertainty surrounding wind speed. This article describes use of a novel Extreme Learning Kalman Filter (ELKF) that integrates the sigma-point Kalman filter with the extreme learning machine algorithm to accurately forecast wind speed sequence using an Artificial Neural Network (ANN)-based state-space model. In the proposed ELKF method, ANNs are used to construct the state equation of the state-space model. The sigma-point Kalman filter is used to address the recursive state estimation problem. Experimental data validations have been implemented to compare the proposed ELKF method with autoregressive (AR) neural networks and ANNs for short-term wind speed forecasting, and the results demonstrated better prediction performance with the proposed ELKF method.

KEYWORDS

wind speed prediction, Kalman filter, uncertain dynamical systems, extreme learning machine, neural networks

1 Introduction

To support environmental sustainability, wind energy is the best green energy source to replace high-carbon power generation. The use of wind power has increased dramatically, particularly in the past decade. It is challenging to control wind turbines and implement optimal wind farm operations for reliable wind power supplementation due to the stochastic nature of wind speed (Evans and Lio, 2022). It is crucial to obtain precise wind speed predictions for optimal control of wind turbines, using methods such as models of predictive control in stabilizing wind turbines (Hur and Leithead, 2022), to ensure stable supply of wind power. Predicting wind speed, especially in the short-term, is critical for improving power generation efficiency and extending the life span of a wind turbine (Bossanyi, 2003; Lio et al., 2017; Liew et al., 2022).

The existing methods for wind speed prediction in the short term represent two different classes. The first class includes physical model-based methods (Cassolaa and Burlando, 2012; Xu et al., 2020; Chen et al., 2021; Li et al., 2022). One physical model-based method is the numerical weather prediction method, in which predictions are based on physical models that include parameters characterizing the properties of the weather, such as atmospheric pressure, surface roughness, temperature, and many others (Lynch, 2008; Chen et al., 2021). However, the performance of the numerical weather prediction method deteriorates dramatically when the uncertainty of weather conditions is high. To address the uncertain nature of wind speed, statistical modeling has become more and more popular compared to models of the physical mechanism of wind speed. In the statistical approach, statistical methods are applied to model the temporal causality of wind speed (Shen et al., 2021; Ouyang et al., 2017). Currently, wind speed prediction often uses time series models or artificial neural network (ANN) models (Giorgi et al., 2011). Among time series model-based methods, autoregressive moving average (ARMA) models, with variants such as simplified autoregressive (AR) models and others, have been well-applied in forecasting wind speed sequence (Prema and Rao, 2015; Torres et al., 2005; Hanoon et al., 2022). However, the ARMA class methods need to increase model order to model causality in a time series, especially considering the stochastic variations of wind speed. Moreover, ARMA models are essentially linear models, which are not able to capture nonlinear features of the causality of wind speed. Therefore, ANNs are used to describe the nonlinear features of temporal causality in wind speed (Başaran and Filik, 2017; Kadhem et al., 2017; Malik et al., 2022). Although ANNs can handle the nonlinearity in wind speed dynamics, they cannot independently capture the dynamical nature of wind speed time series; therefore, a high-order model is necessary, which increases the complexity of the model.

State-space models are naturally suitable for describing the temporal causality of a dynamical system (Shen et al., 2020). Kalman filters can be used to identify the state-space models and infer the hidden state variables (Kitagawa, 1996; Kitagawa, 1987). Kalman filtering has been widely used in estimation problems, such as battery capacity estimation (Plett, 2004a; Plett, 2004b; Plett, 2004c; Plett, 2006a; Plett, 2006b). For nonlinear models, use of a sigma-point Kalman filter (SPKF) is an effective way to conduct the state estimation (Plett, 2006a; Plett, 2006b). However, accuracy of the SPKF relies on the goodness-of-fit of the model. Therefore, in this article, we propose the Extreme Learning Kalman Filter (ELKF), a novel method that combines SPKF and ANN to give accurate short-term predictions of wind speed. Our data indicate that the state-space model for wind speed can be obtained using noisy measurements of wind speed. With an ANN-based state-space model for dynamics of wind speed, we conduct SPKF modeling to obtain estimates of current wind speed, then use the estimation and an ANN-based state equation to further predict wind speed. The effectiveness of the proposed ELKF model has been validated using an experimental data set.

This article is organized into four sections. **Section 2** includes a formal description of the problem. In **Section 3**, the proposed algorithm is presented after a brief introduction to the Extreme Learning Machine (ELM) based training algorithm for ANNs and SPKF. The results of experimental data-based validations are presented in **Section 4**, and the study conclusion appears in **Section 5**.

2 Problem formulation

Let $t = \ 0, 1, 2, \ldots$ define the discrete time index. Let $s_t \in \mathbb{R}$ be the wind speed at t. The following assumption holds throughout this article.

Assumption 1: The stochastic process s_t is a Markov process. Namely, the following holds.

$$\begin{aligned}&\mathbf{Pr}\left\{s_{t+1} = S^{'} | s_t = S, s_{t-1} = S_{t-1}, \ldots, s_0 = S_0\right\}\\ &\quad = \mathbf{Pr}\left\{s_{t+1} = S^{'} | s_t = S\right\}.\end{aligned} \tag{1}$$

Due to **Assumption 1**, we can use the state-space equation to describe the dynamical evolution of s_t, which is written by

$$s_{t+1} = f_{\mathrm{ws}}\left(s_t, \omega_t\right), \tag{2}$$

where $\omega_t \in \mathbb{R}$ is a random variable for system noise, and f_{ws}:$\mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is an unknown nonlinear function that describes the causality relationship between s_t and s_{t+1}. The measurement of wind speed z_t is perturbed by s_t due to the existence of measurement noise v_t. The observation function is thus written by

$$z_t = s_t + v_t. \tag{3}$$

We hold the following assumptions regarding ω_t and v_t.

Assumption 2: The system noise ω_t and measurement noise v_t are both identically independently distributed Gaussian noises.

The problem that needs to be addressed is summarized as follows.

Problem 1. With data set $\mathcal{Z}_t = \{z_0, \ldots, z_t\}$, this study aims to predict $\hat{s}_{t'}$ of $s_{t'}$ for a $t' \ > \ t$. The prediction $\hat{s}_{t'}$ is expected to satisfy

$$\mathbb{E}\left\{\hat{s}_{t'} - s_{t'}\right\} = 0, \tag{4}$$

and

$$\mathbb{E}\left\{\left(\hat{s}_{t'} - s_{t'}\right)^2\right\} = \mathbb{E}\left\{v_t^2\right\}. \tag{5}$$

To achieve the task, the following two sub-problems need to be addressed:

- construction of the approximate model $\hat{f}_{\mathrm{ws}}(\cdot)$ of $f_{\mathrm{ws}}(\cdot)$;
- design of filtering and prediction algorithms to estimate $\hat{s}_t$ and prediction of $\hat{s}_{t'}$.

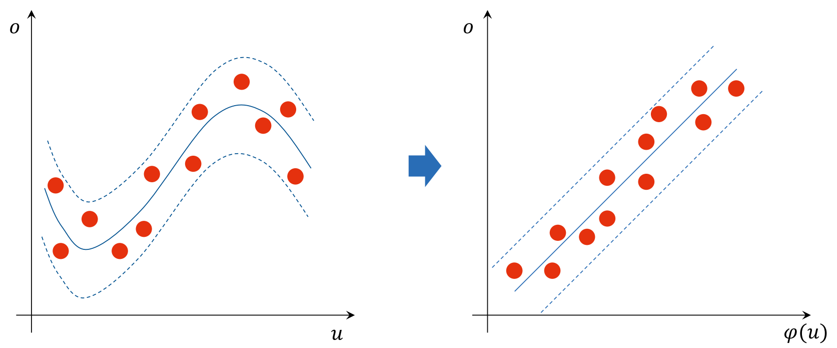

FIGURE 1
Essentials of the ELM algorithm: activation function mapping.

3 Proposed method

3.1 ELM for training ANNs

ELM is an effective algorithm for training single-layer ANNs. Single-layer ANNs can be well trained to approximate standard multilayer ANNs. In a single-layer ANN, there are nonlinear activation functions as hidden nodes. For example, there are N samples (u_i, o_i), where $o_i = [o_{i,1}, \dots, o_{i,m}]^{\mathrm{T}} \in \mathbb{R}^m$ represents the output variable and $u_i = [u_{i,1}, \dots, u_{i,n}]^{\mathrm{T}} \in \mathbb{R}^n$ is the input. A single-layer ANN denoted by $h(u)$ with S activation function $\varphi(\cdot)$ describes the input-to-output causality by

$$\hat{o}_i = h(u_i) == \sum_{j=1}^{S} \beta_j \varphi\left(\omega_j^{\mathrm{T}} u_i + b_j\right),\ i = 1, \dots, N \tag{6}$$

where $\omega_j = [\omega_{j,1}, \dots, \omega_{j,n}]^{\mathrm{T}}$ and $\beta_j = [\beta_{j,1}, \dots, \beta_{j,m}]^{\mathrm{T}}$ are vectors, and b_j is a constant scalar in the jth node. If we consider the relationship between o and u, it is described by $o = h(u)$ and is nonlinear, as shown in the left part of Figure 1. On the other hand, if we consider o and $\varphi(\cdot)$, the causality becomes linear after activation function mapping, as shown in the right part of Figure 1.

A single-layer ANN with S activation function $\varphi(\cdot)$ can approximate these N samples in the sense of zero means written by

$$\sum_{i=1}^{N} e_i = \sum_{i=1}^{N} (o_i - \hat{o}_i) = 0, \tag{7}$$

then, by the theoretical result in Huang (2003), $\exists S \leq N, b_j, \omega_j, \beta_j$ such that

$$o_i = \sum_{j=1}^{S} \beta_j \varphi\left(\omega_j^{\mathrm{T}} u_i + b_j\right),\ i = 1, \dots, N. \tag{8}$$

where the activation function adopts the same function denoted by $\varphi(\cdot)$. More generally, by defining

$$H = \begin{bmatrix} \varphi\left(\omega_1^{\mathrm{T}} u_1 + b_1\right) & \dots & \varphi\left(\omega_{\tilde{N}}^{\mathrm{T}} u_1 + b_{\tilde{N}}\right) \\ \vdots & \dots & \vdots \\ \varphi\left(\omega_{\tilde{N}}^{\mathrm{T}} u_N + b_1\right) & \dots & \varphi\left(\omega_{\tilde{N}}^{\mathrm{T}} u_N + b_{\tilde{N}}\right) \end{bmatrix}, \tag{9}$$

$$\beta = \begin{bmatrix} \beta_1^{\mathrm{T}} \\ \vdots \\ \beta_{\tilde{N}}^{\mathrm{T}} \end{bmatrix}, \tag{10}$$

and

$$O = \begin{bmatrix} o_1^{\mathrm{T}} \\ \vdots \\ o_{\tilde{N}}^{\mathrm{T}} \end{bmatrix}, \tag{11}$$

the above equation is equivalent to

$$O = H\beta. \tag{12}$$

We call H the output matrix of the hidden layer or the hidden layer output matrix. The ith column of H represents the ith hidden node output computed from the inputs $u_1, u_2, \dots, u_N$. Generally, the values of $\beta, \omega_1, \dots, \omega_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}$ can be trained by using a gradient descent algorithm. However, the ELM algorithm proposed in Huang (2003) provides a simpler and more efficient way to obtain good estimations of $\beta, \omega_1, \dots, \omega_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}$ in the sense of zero means.

Let $U = [u_1, u_2, \dots, u_N]$ be the input matrix and O be the output matrix. The ELM algorithm for training single-layer ANNs is summarized in **Algorithm 1**. Note that H^M represents the generalized inverse of H, called the Moore-Penrose generalized inverse, which is defined in Rao and Mitra (1972) and is calculated as

$$H^M = \left(H^{\mathrm{T}} H\right)^{-1} H^{\mathrm{T}}. \tag{13}$$

Thus, we calculate the estimation of β by

$$\beta = \left(H^{\mathrm{T}} H\right)^{-1} H^{\mathrm{T}} O. \tag{14}$$

```
Input: $U,O,S$
Output: $b_j,\omega_j,\beta,j=1,2,\ldots,S$
Step 1: Generate $b_j$ and $\omega_j$, $j=1,\ldots,\bar{N}$ by random algorithm (uniform distribution);
Step 2: Compute $H$;
Step 3: Compute $\beta=H^M O$
```

Algorithm 1. Original ELM for single-layer ANNs.

The effectiveness of Algorithm 1 is theoretically proven in Tamura and Tateishi (1997) when we assume that H is invertible, $\varphi(\cdot)$ is infinitely differentiable, and $\|H\beta-O\|=0$.

A sequential version can also be derived by way of the recursive least squares (RLS) algorithm (Chong and Zak, 2001), which is summarized as

3.2 Brief introduction to SPKF

SPKF is a generalized Kalman filter for estimating the state of nonlinear dynamical systems described by

$$x_{t+1}=f(x_t,w_t), \tag{15}$$

$$y_t=g(x_t,v_t), \tag{16}$$

where $x_t\in\mathbb{R}^{n_x}$ denotes the state variable, $y_t\in\mathbb{R}^m$ is the observation, and w_t,v_t are noises. Instead of using Taylor-series expansion-based approximations of the covariance matrices of the state and output as in the extended Kalman filter (EKF) (Plett, 2004a; Plett, 2004b; Plett, 2004c), several functions are evaluated to calculate the approximations of the covariance matrices. SPKF does not require the differentiability of the original functions in state-space models and can achieve better approximations of covariance matrices, which improves state estimation accuracy. Additionally, in SPKF, the derivatives of the functions in the state-space models are not required, which also reduces the computational complexity of implementing SPKF.

In SPKF, several sigma points are selected to be the input of the nonlinear function. Note that the mean and covariance of the sigma points can be weighted to the values that are equal to those of the *a priori* state estimation. These points are directed to a set of points output by the nonlinear function, which can be used to obtain the approximate mean and covariance of the *a posteriori* estimated state. In the state estimation problem for a system described by (Eqs 15, 16), the required number of sigma points is $p+1=2n_x+1$. The generated set is defined by

$$\mathcal{X}_{t,i}=\left\{\bar{x}_{t,i},\bar{x}_{t,i}+\gamma\sqrt{\Sigma_{x_t}},\bar{x}_{t,i}-\gamma\sqrt{\Sigma_{x_t}}\right\},\quad i=0,\ldots,p, \tag{17}$$

where the matrix square root $B=\sqrt{A}$ is the result of $A=BB^{\mathrm{T}}$, $\bar{x}_t$ is the mean of x_t, and Σ_{x_t} is the covariance of x_t. It can find a specific set of $\{\gamma,\alpha^{(m)},\alpha^{(c)}\}$ such that the weighted mean and covariance of $\mathcal{X}_{t,i}$ equal the original mean and covariance of x_t, which are written by

$$\bar{x}=\sum_{i=0}^{p}\alpha_i^{(m)}\mathcal{X}_{t,i}, \tag{18}$$

$$\Sigma_{x_t}=\sum_{i=0}^{p}\alpha_i^{(c)}\left(\mathcal{X}_{t,i}-\bar{x}\right)\left(\mathcal{X}_{t,i}-\bar{x}\right)^{\mathrm{T}}. \tag{19}$$

Note that $\sum_{i=0}^{p}\alpha_i^{(m)}=1$ and $\sum_{i=0}^{p}\alpha_i^{(c)}=1$.

Let $x_t^a=[(x_t)^{\mathrm{T}},\ w_t,v_t]^{\mathrm{T}}$ be an augmented random variable. With x_t^a, we implemented SPKF as follows.

1) Update of the *a priori* state estimation. First, there is augmentation of *a posteriori* state estimation of the last time step,

$$\hat{x}_{t-1}^{a,+}=\left[\left(\hat{x}_t^{+}\right)^{\mathrm{T}},\ \bar{w}_t,\bar{v}_t\right]^{\mathrm{T}}, \tag{20}$$

and then augmentation of *a posteriori* covariance estimation,

$$\Sigma_{x_{t-1}}^{a,+}=\operatorname{diag}\left(\Sigma_{x_{t-1}}^{+},\Sigma_w,\Sigma_v\right). \tag{21}$$

Based on (Eqs 20, 21), we can generate $p+1$ sigma points

$$\mathcal{X}_{t-1,i}^{a,+}=\left\{\bar{x}_{t-1,i}^{a,+},\bar{x}_{t-1,i}^{a,+}+\gamma\sqrt{\Sigma_{x_{t-1}}^{a,+}},\bar{x}_{t-1,i}^{a,+}-\gamma\sqrt{\Sigma_{x_{t-1}}^{a,+}}\right\},i=0,\ldots,p. \tag{22}$$

In $\mathcal{X}_{t-1,i}^{a,+}$, we extract the state portion $\mathcal{X}_{t-1,i}^{x,+}$ and the measurement noise portion $\mathcal{X}_{t-1,i}^{w,+}$ to evaluate the state equation to calculate the *a priori* sigma points $\mathcal{X}_{t,i}^{x,-}$ for time step t, which is written as

$$\mathcal{X}_{t,i}^{x,-}=f\left(\mathcal{X}_{t-1,i}^{a,+},\mathcal{X}_{t-1,i}^{w,+}\right). \tag{23}$$

The *a priori* state estimation at time step t is calculated as

$$\begin{aligned}\hat{x}_t^{-}&=\mathbb{E}\{f(x_{t-1},w_{t-1})\,|\,\mathbb{Y}_{t-1}\},\\&\approx\sum_{i=0}^{p}\alpha_i^{(m)}f\left(\mathcal{X}_{t-1,i}^{a,+},\mathcal{X}_{t-1,i}^{w,+}\right)=\sum_{i=0}^{p}\alpha_i^{(m)}\mathcal{X}_{t,i}^{x,-}.\end{aligned} \tag{24}$$

2) Update of the error covariance. The *a priori* covariance estimation is obtained from the *a priori* sigma points $\mathcal{X}_{t,i}^{x,-},\quad i=0,1,\ldots,p$. That is

$$\Sigma_{x_t}^{-}=\sum_{i=0}^{p}\alpha_i^{(c)}\left(\mathcal{X}_{t,i}^{x,-}-\hat{x}_t^{-}\right)\left(\mathcal{X}_{t,i}^{x,-}-\hat{x}_t^{-}\right)^{\mathrm{T}}. \tag{25}$$

3) Estimation of the system output. The estimation of system output is computed from sigma points by using the observation equation. First, we calculate the estimated points

$$\mathcal{Y}_{t,i}=g\left(\mathcal{X}_{t,i}^{x,-},\mathcal{X}_{t,i}^{v,-}\right). \tag{26}$$

The output prediction is then given by

$$\begin{aligned}\hat{y}_t&=\mathbb{E}\{h(x_t,v_t)\,|\,\mathbb{Y}_{t-1}\},\\&\approx\sum_{i}^{(m)}h\left(\mathcal{X}_{t,i}^{x,-},\mathcal{X}_{t,i}^{v,-}\right)=\sum_{i=0}^{p}\alpha_i^{(m)}\mathcal{Y}_{t,i}.\end{aligned} \tag{27}$$

```
TRAIN($\hat{f}_{\mathrm{ws}}(\cdot)$)
    Measure the data set $\mathcal{Z}_T$
    choose activation function $\phi(\cdot)$
    set the number of activation functions $S$
    Implement Algorithm 1 and obtain $\hat{f}_{\mathrm{ws}}(\cdot)$
    return $\hat{f}_{\mathrm{ws}}(\cdot)$
PREDICT($\hat{s}_{t'}$)
    Update $\hat{f}_{\mathrm{ws}}(\cdot)$ with new data $z_t$ via Algorithm 2
    Implement SPKF to obtain $\hat{s}_t^+$ based on $\hat{f}_{\mathrm{ws}}(\cdot)$
    Calculate $\hat{s}_{t'}^-$ based on $\hat{f}_{\mathrm{ws}}(\cdot)$ and $\hat{s}_t^+$
    return $\hat{s}_{t'}^-$
```

Algorithm 2. Algorithm for the proposed ELKF method.

```
Step 1: Initialize $\beta$ by $\beta_0$ which is obtained
from history data by Algorithm 1
Step 2: Compute $h_{t+1}$ based on new measurement
$(u_{t+1},o_{t+1})$ according to Eq (9), $k=0,1,2,\ldots,i,\ldots$
Step 3: Update $\beta_{t+1}$ by

        $\beta_{t+1}=\beta_t+M_{t+1}h_{t+1}(o_{t+1}^{\mathrm{T}}-h_{t+1}^{\mathrm{T}}\beta_t)$        (33)

where $M_{t+1}$ is computed by

        $M_{t+1}=M_t-\frac{M_k h_{t+1}h_{t+1}^{\mathrm{T}}M_t}{1+h_{t+1}^{\mathrm{T}}M_t h_{t+1}}.$        (34)

Step 4: Set $t=t+1$
```

Algorithm 3. Sequential ELM algorithm.

4) Determination of the estimated gain matrix. To calculate the estimated gain matrix, the required covariance matrices are first be computed by

$$\Sigma_{y_t}=\Sigma_{i=0}^{p}\alpha_i^{(c)}\left(\mathcal{Y}_{t,i}-\hat{y}_t\right)\left(\mathcal{Y}_{t,i}-\hat{y}_t\right)^{\mathrm{T}}, \tag{28}$$

$$\Sigma_{x_t,y_t}^{-}=\Sigma_{i=0}^{p}\alpha_i^{(c)}\left(\mathcal{X}_{t,i}^{x,-}-\hat{x}_t^{-}\right)\left(\mathcal{Y}_{t,i}-\hat{y}_t\right)^{\mathrm{T}}. \tag{29}$$

Then, we simply compute $L_t=\Sigma_{x_t,y_t}^{-}\Sigma_{y_t}^{-}$.

5) Update of the *a posteriori* state estimation. The *a posteriori* state estimation is

$$\hat{x}_t^{+}=\hat{x}_t^{-}+L_t\left(y_t-\hat{y}_t\right). \tag{30}$$

6) Computation of error covariance. We compute the covariance matrix of error by

$$\Sigma_{x_t}^{+}=\Sigma_{x_t}^{-}-L_t\Sigma_{\hat{y}_t}L_t^{\mathrm{T}}. \tag{31}$$

3.3 Proposed algorithm

The framework of the ELKF method for predicting wind speed is illustrated in Figure 2 and the proposed algorithm is summarized in Algorithm 2.

In the training step, the history data set $\mathcal{Z}_T$ is used to train the approximate model $\hat{f}_{\mathrm{ws}}(\cdot)$ of $f_{\mathrm{ws}}(\cdot)$. The model is written by

$$z_t=\hat{f}_{\mathrm{ws}}\left(z_{t-1}\right)=\sum_{j=1}^{S}\beta_j\varphi\left(\omega_j^{\mathrm{T}}z_{t-1}+b_j\right),\ t=1,\ldots,T. \tag{32}$$

By implementing Algorithm 1, we obtain $\beta_j,\omega_j^{\mathrm{T}},b_j$ for $j=1,\ldots,S$ with data set $\mathcal{Z}_T$. Here, z_t is used instead of s_t. The following theorem states that we can use z_t to get $\hat{f}_{\mathrm{ws}}(\cdot)$ in the sense of mean value.

Theorem 1. Assume that Assumption 2 holds. Then, if

$$\mathbb{E}\{z_t\}=\mathbb{E}\left\{\hat{f}_{\mathrm{ws}}\left(z_{t-1}\right)\right\}\ \ t=1,\ldots,T, \tag{35}$$

then the following equation holds

$$\mathbb{E}\{s_t\}=\mathbb{E}\left\{\hat{f}_{\mathrm{ws}}\left(s_{t-1}\right)\right\}\ \ t=1,\ldots,T. \tag{36}$$

Proof. Since Assumption 2 holds. We have

$$\begin{aligned}\mathbb{E}\{z_t\}&=\mathbb{E}\{s_t+v_t\},\\&=\mathbb{E}\{s_t\}+\mathbb{E}\{v_t\},\\&=\mathbb{E}\{s_t\}.\end{aligned} \tag{37}$$

On the other hand,

$$\begin{aligned}\mathbb{E}\left\{\hat{f}_{\mathrm{ws}}\left(z_{t-1}\right)\right\}&=\hat{f}_{\mathrm{ws}}\left(\mathbb{E}\{z_{t-1}\}\right),\\&=\hat{f}_{\mathrm{ws}}\left(\mathbb{E}\{s_{t-1}+v_{t-1}\}\right),\\&=\hat{f}_{\mathrm{ws}}\left(\mathbb{E}\{s_{t-1}\}+\mathbb{E}\{v_{t-1}\}\right),\\&=\hat{f}_{\mathrm{ws}}\left(\mathbb{E}\{s_{t-1}\}\right),\\&=\mathbb{E}\left\{\hat{f}_{\mathrm{ws}}\left(s_{t-1}\right)\right\}.\end{aligned} \tag{38}$$

Thus, (36) holds.

In the implementation step, the estimated model $\hat{f}_{\mathrm{ws}}(\cdot)$ is updated with online data z_t by Algorithm 3. Then, with $\hat{f}_{\mathrm{ws}}(\cdot)$, SPKF is implemented to obtain the *a posteriori* state estimation $\hat{s}_t^{+}$. With $\hat{s}_t^{+}$, the prediction at time step $t' > t$ is calculated by iteratively implementing the following equation

$$\hat{s}_{t_j+1}^{-}=\hat{f}_{\mathrm{ws}}\left(\hat{s}_{t_j}^{-}\right),\ j=0,\ldots,\left(t'-t\right), \tag{39}$$

where $t_0=\ t$.

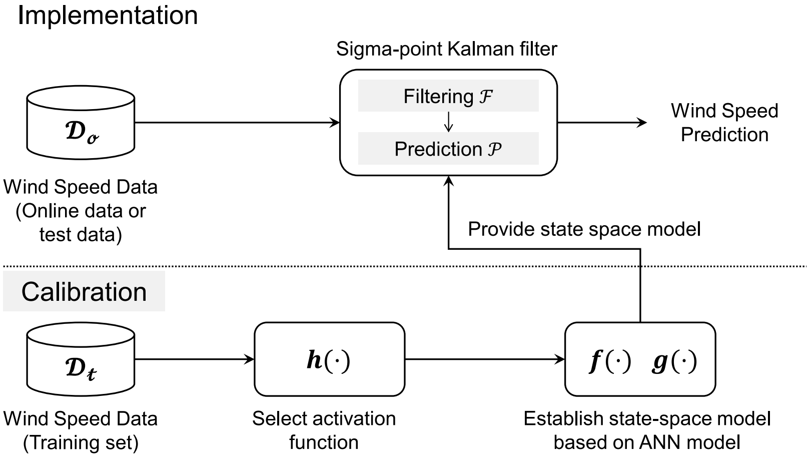
Implementation
Sigma-point Kalman filter
Filtering $\mathcal{F}$
Prediction $\mathcal{P}$
$\mathcal{D}_o$
Wind Speed Data (Online data or test data)
Wind Speed Prediction
Provide state space model
Calibration
$\mathcal{D}_t$
Wind Speed Data (Training set)
$h(\cdot)$
Select activation function
$f(\cdot)$ $g(\cdot)$
Establish state-space model based on ANN model

FIGURE 2
Framework of the ELKF method for predicting wind speed.

4 Validation

4.1 Methods for comparison and the wind speed data set

Figure 3 illustrates the data set used in validation. The data sampling time is 10 min. A total of 61,858 data points were collected from a large wind farm located in Jiugongshan, Hubei, China within 12 consecutive days. The wind speed profiles exhibit similarity in time scale. We used the data from Day 1 to Day 10 for training the model and the rest of the data as the test set.

We compared the proposed ELKF method with the AR model and the ANN model. The ANN model is as described in Section 3. For n-step-ahead prediction, the addressed output is $z_{t'}$ and the input is $[z_{t'-n}, z_{t'-n-1}, \ldots, z_{t'-n-p+1}]^{\mathrm{T}}$. Here, p is the order. The regular AR model is expressed by

$$z_{t'} = \alpha_1 z_{t'-n} + \cdots + \alpha_p z_{t'-n-p+1}, \tag{40}$$

where $\alpha_1, \ldots, \alpha_p$ are regression coefficients.

4.2 Results and discussion

Figure 4 and Figure 5 show plots of the results of predictions (one-step and five-step) for 200 samples using the AR model, ANN model, and ELKF method. The proposed ELKF method exhibits better performance compared to the AR model and the ANN model for both prediction horizons, since the predictions given by the AR and ANN models deviate significantly from the actual values. The predictions given by the proposed ELKF method are close to the actual values. Additionally, histograms of prediction residuals for both prediction horizons are shown in Figure 6. The proposed method has the predicted residuals concentrated to zero with relatively small fluctuation. The traditional methods result in residuals with larger variation. Therefore, our data suggest that the proposed ELKF method provides more reliable predictions.

Furthermore, the correlations between the predictions and the actual values of the samples in Figures 4 and 5 are plotted in Figure 7 and Figure 8, respectively. Compared to the AR and ANN models, the predictions given by the proposed ELKF

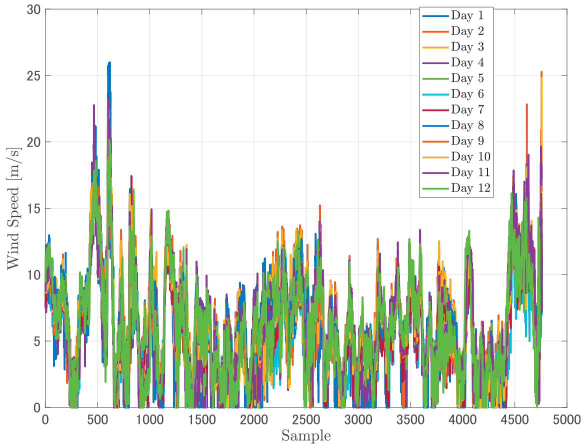

FIGURE 3
Data set for validation.

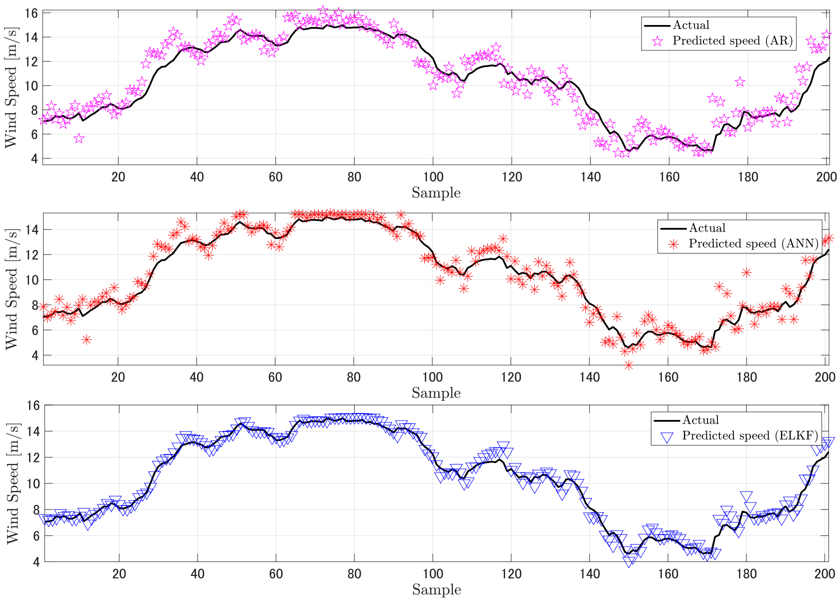

FIGURE 4
Results of wind speed prediction (predictive horizon: one-step): AR, ANN, and ELKF.

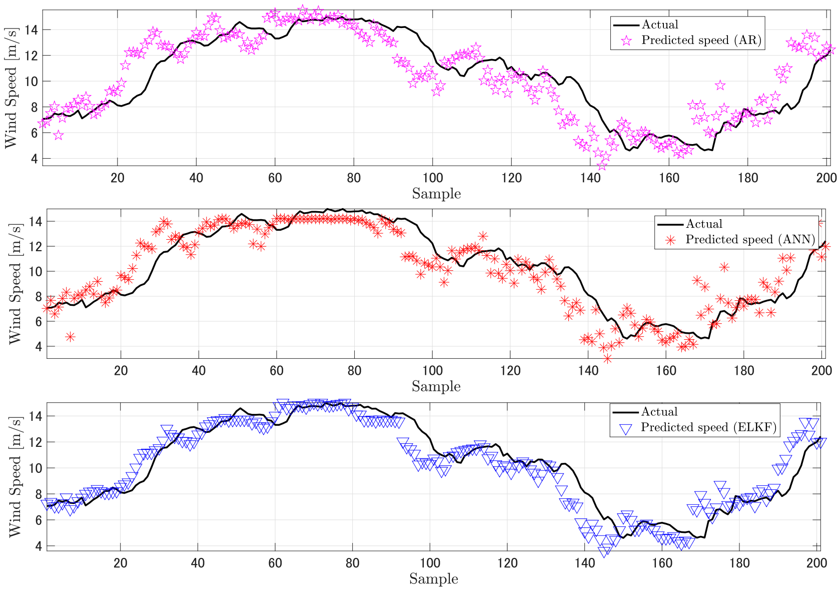

FIGURE 5
Results of wind speed prediction (predictive horizon: five-step): AR, ANN, and ELKF.

method show smaller deviations and strong correlations between the actual values and predictions.

The results of root mean square error (RMSE) of performance of different methods from analysis of the full test data are summarized in Table 1. The results of the mean absolute error (MAE) calculation based on the performance of different methods are summarized in Table 2. RMSE is calculated by

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}. \tag{41}$$

MAE is calculated by

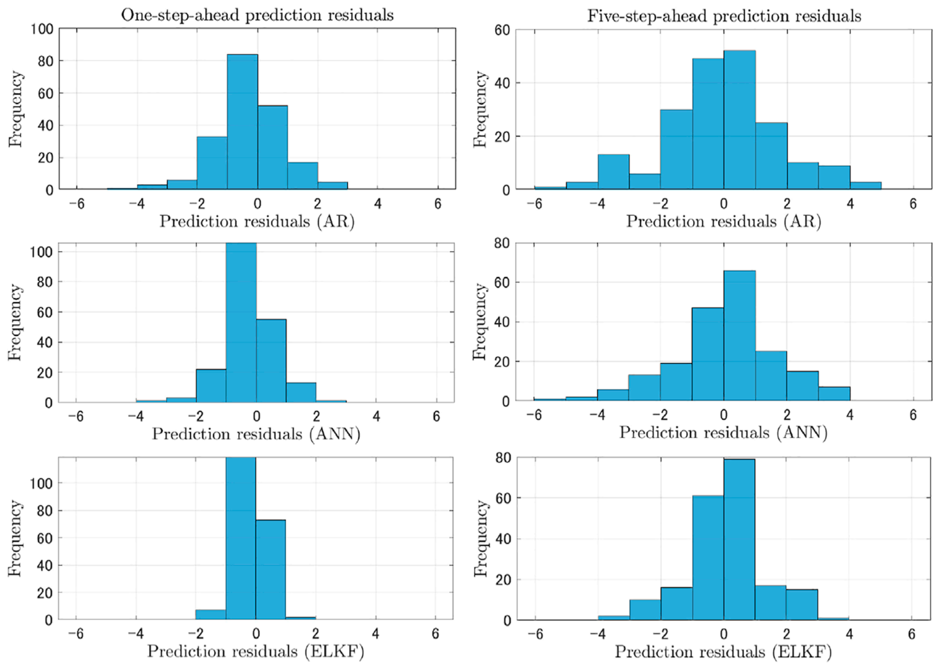

FIGURE 6
Probabilistic histograms of prediction residuals of AR, ANN, and ELKF.

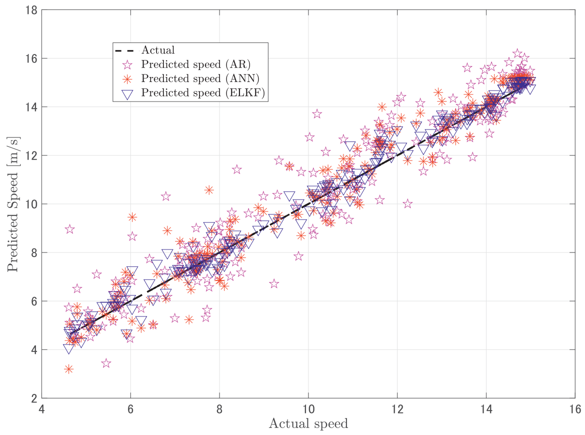

FIGURE 7
Correlation plots of one-step-ahead predictions *versus* actual values: AR, ANN, and ELKF.

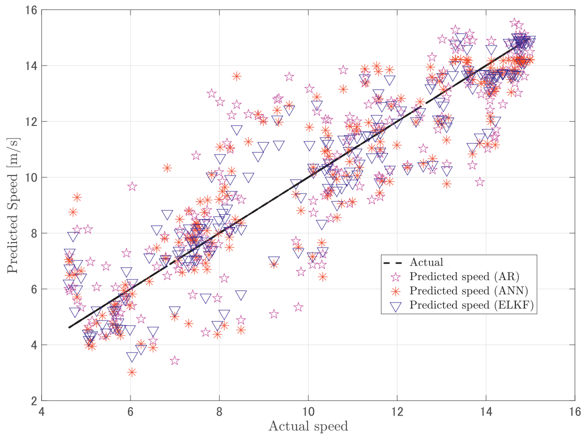

FIGURE 8
Correlation plots of five-steps-ahead predictions *versus* actual values: AR, ANN, and ELKF.

$$\mathrm{MAE} = \frac{1}{T}\sum_{t=1}^{T}|y_t - \hat{y}_t|. \tag{42}$$

Our data indicate that the proposed ELKF method outperforms the AR and ANN models in different predictive horizons. The prediction performance of all methods degrades when the prediction horizon increases because the stochasticity become stronger when the prediction horizon becomes longer.

The reason that the AR model does not give good predictions is that it is a linear time-series model that does not account for dynamic state update in wind speed sequence. The ANN adopts a nonlinear model and is able to depict the nonlinear

TABLE 1 Comparison of RMSE values (m/s) associated with values predicted by AR, ANN, and ELKF.

Predicted horizon	AR	ANN	ELKF
One-step-ahead	0.9918	0.7660	0.4210
Two-steps-ahead	1.1291	0.8936	0.5312
Three-steps-ahead	1.3076	1.1729	0.7819
Four-steps-ahead	1.4375	1.3393	0.9765
Five-steps-ahead	1.7627	1.5864	1.2271

TABLE 2 Comparison of MAE values (m/s) associated with values predicted by AR, ANN, and ELKF.

Predicted horizon	AR	ANN	ELKF
One-step-ahead	0.7491	0.5855	0.3267
Two-steps-ahead	0.8583	0.6878	0.4161
Three-steps-ahead	1.0054	0.9129	0.6096
Four-steps-ahead	1.1086	1.0493	0.7623
Five-steps-ahead	1.3823	1.2719	0.9589

TABLE 3 Comparison of AR, ANN, and ELKF computational loads.

Items	AR	ANN	ELKF
Training time (s)	5.819	93.452	10.521
Online implementation (ms)	0.87	1.91	1.83

feature of causality in wind speed sequence. However, the feature of dynamic state update remains unresolved. The proposed ELKF method adopts an ANN model in the dynamic system to address the nonlinear issue and uses SPKF to resolve the feature of dynamic state update. Therefore, the proposed ELKF method provides better performance in wind speed prediction.

Table 3 summarizes and compares the computation loads of the three methods. The online implementation time adopts the mean computation time in every step. The proposed ELKF has a slightly heavier computational load than AR in both the training process and online implementation. When using an ANN, the computation load dramatically increases during the training process. This is due to adoption of the ELM algorithm by the ELKF to train the single-layer ANN, which is more efficient.

5 Conclusion

In this article, a novel ELKF method is presented for predicting wind speed in the short term. The ELKF method combines the state-space model integrated with the ANN model and state estimation by SPKF. The ANN model is trained by the ELM algorithm and can be updated by a sequential ELM algorithm, which describes the nonlinearity of temporal causality in the time series data. Additionally, by using SPKF for state estimation, the proposed method can capture the dynamical feature of state updates in wind speed time series data. The proposed method can handle the high level of uncertainty of wind speed and produce better predictions compared to the traditional methods. Future work will focus on investigating wind power prediction for various wind turbines and in development of control methods for optimizing wind farm operation. For the proposed ELKF method, all collected data are assumed to be normal data. However, abnormal data may exist in collected data sets. Future work will also focus on using the clustering method to clean data sets to further improve wind speed prediction accuracy.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material. Further inquiries can be directed to the corresponding author.

Author contributions

The author confirms being the sole contributor to this work and has approved it for publication.

Acknowledgments

The author thanks Nan Yang at China Three Gorges University for his assistance in providing the experimental data set and his helpful comments that greatly improved the manuscript.

Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Başaran, U., and Filik, F. (2017). Wind speed prediction using artificial neural networks based on multiple local measurements in eskisehir. *Energy Procedia* 107, 264–269. doi:10.1016/j.egypro.2016.12.147

Bossanyi, E. (2003). Wind turbine control for load reduction. *Wind Energy (Chichester).* 6, 229–244. doi:10.1002/we.95

Cassolaa, F., and Burlando, M. (2012). Wind speed and wind energy forecast through kalman filtering of numerical weather prediction model output. *Appl. Energy* 99, 154–166. doi:10.1016/j.apenergy.2012.03.054

Chen, H., Birkelund, Y., Anfinsen, S., Staupe-Delgado, R., and Yuan, F. (2021). Assessing probabilistic modelling for wind speed from numerical weather prediction model and observation in the arctic. *Sci. Rep.* 11, 7613. doi:10.1038/s41598-021-87299-4

Chong, E., and Zak, S. (2001). *An introduction to optimization*. New York: John Wiley.

Evans, M., and Lio, W. (2022). Computationally efficient model predictive control of complex wind turbine models. *Wind Energy* 25, 735–746. doi:10.1002/we.2695

Giorgi, M. D., Ficarella, A., and Tarantino, M. (2011). Error analysis of short term wind power prediction models. *Appl. Energy* 88, 1298–1311. doi:10.1016/j.apenergy.2010.10.035

Hanoon, M., Ahmed, A. N., Kumar, P., Razzaq, A., Zaini, N., Huang, Y. F., et al. (2022). Wind speed prediction over Malaysia using various machine learning models: Potential renewable energy source. *Eng. Appl. Comput. Fluid Mech.* 16, 1673–1689. doi:10.1080/19942060.2022.2103588

Huang, G. (2003). Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Trans. Neural Netw.* 14, 274–281. doi:10.1109/TNN.2003.809401

Hur, S., and Leithead, W. (2022). Model predictive and linear quadratic Gaussian control of a wind turbine. *Optim. Control Appl. Methods* 38, 88–111. doi:10.1002/oca.2244

Kadhem, A., Wahab, N., Aris, I., Jasni, J., and Abdalla, A. (2017). Advanced wind speed prediction model based on a combination of weibull distribution and an artificial neural network. *Energies* 10, 1744. doi:10.3390/en10111744

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statistics* 5, 1–25. doi:10.2307/1390750

Kitagawa, G. (1987). Non-Gaussian state-space modeling of nonstationary time series. *J. Am. Stat. Assoc.* 82, 1032–1063. doi:10.2307/2289375

Li, Y., Tang, F., Gao, X., Zhang, T., Qi, J., Xie, J., et al. (2022). Numerical weather prediction correction strategy for short-term wind power forecasting based on bidirectional gated recurrent unit and xgboost. *Front. Energy Res.* 9. doi:10.3389/fenrg.2021.836144

Liew, J., Göçmen, T., Lio, W., and Larsen, G. (2022). Streaming dynamic mode decomposition for short-term forecasting in wind farms. *Wind Energy* 25, 719–734. doi:10.1002/we.2694

Lio, W., Jones, B., and Rossiter, J. A. (2017). Preview predictive control layer design based upon known wind turbine blade-pitch controllers. *Wind Energy (Chichester).* 20, 1207–1226. doi:10.1002/we.2090

Lynch, P. (2008). The origins of computer weather prediction and climate modeling. *J. Comput. Phys.* 227, 3431–3444. doi:10.1016/j.jcp.2007.02.034

Malik, P., Gehlot, A., Singh, R., Gupta, L. R., and Thakur, A. K. (2022). A review on ann based model for solar radiation and wind speed prediction with real-time data. *Arch. Comput. Methods Eng.* 29, 3183–3201. doi:10.1007/s11831-021-09687-3

Ouyang, T., Kusiak, A., and He, Y. (2017). Modeling wind-turbine power curve: A data partitioning and mining approach. *Renew. Energy* 102, 1–8. doi:10.1016/j.renene.2016.10.032

Plett, G. (2004a). Extended kalman filtering for battery management systems of lipb-based hev battery packs-part 1: Background. *J. Power Sources* 134, 252–261. doi:10.1016/j.jpowsour.2004.02.031

Plett, G. (2004b). Extended kalman filtering for battery management systems of lipb-based hev battery packs-part 2: Modeling and identification. *J. Power Sources* 134, 262–276. doi:10.1016/j.jpowsour.2004.02.032

Plett, G. (2004c). Extended kalman filtering for battery management systems of lipb-based hev battery packs-part 3: State and parameter estimation. *J. Power Sources* 134, 277–292. doi:10.1016/j.jpowsour.2004.02.033

Plett, G. (2006a). Sigma-point kalman filtering for battery management systems of lipb-based hev battery packs: Part 1: Introduction and state estimation. *J. Power Sources* 161, 1356–1368. doi:10.1016/j.jpowsour.2006.06.003

Plett, G. (2006b). Sigma-point kalman filtering for battery management systems of lipb-based hev battery packs: Part 2: Simultaneous state and parameter estimation. *J. Power Sources* 161, 1369–1384. doi:10.1016/j.jpowsour.2006.06.004

Prema, V., and Rao, K. U. (2015). Time series decomposition model for accurate wind speed forecast. *Renewables.* 2, 18. doi:10.1186/s40807-015-0018-9

Rao, C., and Mitra, S. (1972). *Generalized inverse of matrices and its application*. New York: Wiley.

Shen, X., Ouyang, T., Yang, N., and Zhuang, J. (2021). Sample-based neural approximation approach for probabilistic constrained programs. *IEEE Trans. Neural Netw. Learn. Syst.* 34 (2), 1058–1065. doi:10.1109/TNNLS.2021.3102323

Shen, X., Ouyang, T., Zhang, Y., and Zhang, X. (2020). Computing probabilistic bounds on state trajectories for uncertain systems. *IEEE Trans. Emerg. Top. Comput. Intell.* 7 (1), 285–290. doi:10.1109/TETCI.2020.3019040

Tamura, S., and Tateishi, M. (1997). Capabilities of a four-layered feedforward neural network: Four layers versus three. *IEEE Trans. Neural Netw.* 8, 251–255. doi:10.1109/72.557662

Torres, J. L., Garcia, A., De Blas, M., and De Francisco, A. (2005). Forecast of hourly average wind speed with arma models in navarre (Spain). *Sol. Energy* 79, 65–77. doi:10.1016/j.solener.2004.09.013

Xu, W., Ning, L., and Luo, Y. (2020). Wind speed forecast based on post-processing of numerical weather predictions using a gradient boosting decision tree algorithm. *Atmosphere* 11, 738. doi:10.3390/atmos11070738