Wind speed forecast based on combined theory, multi-objective optimisation, and sub-model selection

Tonglin Fu1 · Shenghui Zhang2

Accepted: 22 June 2022 / Published online: 23 August 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract
Wind energy is the primary energy source for a sustainable and pollution-free global power supply. However, because of its characteristic irregularity, nonlinearity, non-stationarity, randomness, and intermittency, previous studies have only focused on stability or accuracy, and the forecast performances of their models were poor. Moreover, in previous research, the selection of sub-models used for the combined model was not considered, which weakened the generalisability. Therefore, to further improve the forecast accuracy and stability of the wind speed forecasting model, and to solve the problem of sub-model selection in the combined model, this study developed a wind speed forecasting model using data pre-processing, a multi-objective optimisation algorithm, and sub-model selection for the combined model. Simulation experiments showed that our combined model not only improved the forecasting accuracy and stability but also chose different sub-models and different weights of the combined model for different data; this improved the model generalisability. Specifically, the MAPEs of our model are less than 4.96%, 4.60%, and 5.25% in one-, two-, and three-step forecast. Thus, the proposed combined model is demonstrated as an effective tool for grid dispatching.

Keywords Wind energy · Wind speed forecasting · Sub-model selection · Combined model · Multi-objective optimisation

1 Introduction
For decades, the demand for electricity has represented a major global constraint, and the use of renewable energy has become increasingly important. From the perspective of economic growth, energy plays a vital role in procuring power from nature. Fossil fuels have been used to generate electricity; however, owing to the related fossil fuel crisis and current global environmental issues, the energy mixture is changing. Renewable energy sources (e.g. wind, tidal, and solar energy) are gaining increasing amounts of attention (Wang et al. 2004). Wind energy is clean, abundant, environmentally friendly, inexhaustible, and inexpensive; it represents a feasible alternative to fossil fuels.

In recent decades, significant progress has been made in wind turbines, from the small to large scales. The reason for this rapid growth is that many parts of the world are rich in the required raw materials, the turbines are ecologically friendly and achieve low-carbon power generation, and new policies have been implemented towards the introduction of new renewable generators (Enrique et al. 2019; Aguilar Vargas et al. 2019). For example, in France, wind power installation increased by 14.04% (Council and Global wind statistics 2019) in 2017, mainly owing to tariff subsidies. According to statistical research, the global cumulative installed wind capacity reached nearly 591 Gigawatts (GW) by the end of 2018, with an annual growth rate of 9.6% (Fried et al. 2021). However, owing to changes in wind speed and direction (especially the former), the integration of a large portion of wind power in wind power systems faces major challenges (Meng et al. 2016). These challenges can be classified into two categories: operational issues (Yan et al. 2017) and planning
and economic issues (Bruninx et al. 2016). Improving wind forecasting is one of the most effective ways to overcome these challenges.

In recent research, improving the forecast accuracy of wind speed has become a hot topic, and new development directions and numerous wind speed forecasting methods have been proposed. Some of these technologies provide more accurate wind speed forecasts for specific wind speed data, whereas others are effective for multiple wind speed dataset (Chang 2014). These different wind-speed forecasting models can be classified into the following six categories:

1. **Persistence models.** Persistence models, which assume that the wind speed at a certain time in the future matches the forecasting wind speed (Soman et al. 2010), are used for short-term forecasts. However, when the forecast timescale increases, the forecast accuracy of the model decreases. These models can be used as reference models to test new models for short-term wind-speed forecasts (Sfetsos 2000).

2. **Physical models.** Physical models are used in atmospheric weather forecast modelling; they require copious amounts of numerical weather forecast data, including humidity, temperature, pressure, speed, and topological parameters; this leads to data accumulation. To forecast the wind speed, a long calculation time is required for data correlation. Therefore, the model is best suited for long-term wind speed forecasting (Lei et al. 2009).

3. **Statistical models.** Historical data are used in statistical models to forecast the wind speed. Statistical models include linear and nonlinear models, and these have also been used in time series forecasting (Smith and Mehta 1993; Riahy and Abedi 2008). However, for linear methods such as the auto-regressive moving average (Torres et al. 2005; Brown et al. 1984), auto-regressive integrated moving average (ARIMA) (Grigonte and Butkevichiute 2016), Box–Jenkins method, and Markov chain models (Shamshad et al. 2005), when the main component of the wind speed data set is a nonlinear feature, most statistical models assume that wind speed to be normally distributed; hence, the accuracy of these forecasting methods decreases rapidly. These nonlinear statistical methods require a large amount of historical data to train and develop forecasts; they primarily include fuzzy logic methods (Damousis et al. 2004), support vector machines (SVMs) (Zhang et al. 2017; Niu et al. 2018), and probabilistic methods (Heng et al. 2022).

4. **Hybrid models.** Hybrid methods combine complementary characteristics (e.g. linear and nonlinear data) and the advantages of various methods to obtain the best forecasting performance.

5. **Artificial intelligence methods.** Artificial intelligence methods, including artificial neural networks (ANNs) such as the back propagation neural network (BPNN) (Guo et al. 2011), generalised regression neural network (GRNN) (Specht 1991), radial basis function neural network (RBFNN) (Specht 1991), multiple layer perceptron (Wang et al. 2014), and long short-term memory (LSTM) (Dorvlo et al. 2002), have also been proposed to forecast wind speeds (Li and Shi 2010; Yu et al. 2017; Wang et al. 2016). To improve the forecasting accuracy, researchers have also used optimisation algorithms such as differential evolution (Yang et al. 2016) and cuckoo search (Xiao et al. 2015), amongst others.

6. **Combined models.** Depending on the above methods, the stability and forecasting accuracy of forecasting beyond these methods do not meet the levels desired by wind farm operators. Therefore, to achieve a higher level of accuracy and more stable forecasting results, researchers have developed a combination model that incorporates the advantages of a single forecast model and can be widely used for wind-speed forecasting (Zhang et al. 2022; Wang et al. 2021; Xiao et al. 2018).

In general, the combined models not only overcome certain difficulties of the single models but also combine the advantages of these models, making them better than the models mentioned above. However, for combined models, the selection of sub-models has always been difficult. Some researchers choose linear and nonlinear models so that the combined models can satisfy both linear and nonlinear data. Some researchers have selected models that perform better on a given dataset.

However, certain problems remain: (a) The lack of complementary theoretical knowledge for selecting linear and nonlinear models; in other words, why are these methods selected as sub-models? (b) The listed sub-models are limited. For the resulting combined model, the sub-models represent the best of these models. Because the sub-models do not cover most models, the single model selected is not sufficiently convincing. (c) For different datasets, because of the different data characteristics, it is impossible to find a single combined model suitable for multiple datasets. (d) Most of the models feature single objectives. For wind speed data, it is usually difficult to guarantee both forecasting accuracy and stability using a single objective function in the combination model, owing
to the nonlinear characteristics of the wind-speed time series.

To the best of our knowledge, no model has been proposed that can resolve the problems stated above. Thus, we propose a forecasting model based on model selection, multi-objective functions, and combined model theory. In the following section, we use CM to denote our proposed combined model.

Our contributions and innovations are as follows:

1. A combined model based on outlier detection and processing is constructed, and the negative effects of outliers are eliminated using data analysis methods, whilst retaining the main trend of the wind-speed time series. Outliers in the original wind-speed data can result in poor forecasting results, and the data analysis module of the CM eliminates such outliers.

2. Based on singular spectrum analysis (SSA) technology, data pre-processing technology is used to extract the main features of the wind speed data. The wind speed data are denoised to make them smoother and more reflective of the trend of the original data.

3. An optimal sub-model selection criterion based on multiple forecast criteria is proposed. A new predictive evolution criterion is proposed to select the optimal predictive sub-model. This criterion is called the weighted information criterion (WIC), and it combines six criteria [mean absolute percentage error (MAPE), root mean square error (RMSE), Akaike information criterion (AIC), Bayesian information criterion (BIC), and direction accuracy (DA)]. These criteria are used not only to select the best sub-models from various forecasting models but also to improve the forecasting accuracy.

4. The weights of the sub-models used to build the combined model are obtained using a multi-objective optimisation algorithm.

5. The single model and weight of the ideal combination model vary with respect to the data, indicating the lack of a consistent model suitable for all datasets.

The remainder of this paper is organised as follows. The theories and methods are introduced in Sect. 2. In Sect. 3, we introduce the multiple objective functions, non-dominated sorting genetic algorithm-III, and model-building process. In Sect. 4, the performance metrics, three numerical experiments, and summary are presented. Finally, Sect. 5 presents the conclusions.

2 Theories and methods

This section introduces the theories and methods used in CM. In the attached table in the appendix, we fit the original wind-speed time series using linear and nonlinear functions; we found that the wind speed had both linear and nonlinear characteristics. Therefore, linear and nonlinear models were selected to forecast and study the wind-speed time series. The basic methods and theories were shown in the appendix. The flowchart of the proposed combined model is shown in Fig. 1.

3 The non-dominated sorting genetic Algorithm-III and model proposal

For a single-objective model, only one objective function must be determined for forecasting. However, theoretical and practical examples have shown that it is difficult to achieve high stability and accuracy by relying on only one objective. For multi-objective problems, multiple goals must be proposed for the optimisation algorithm to optimise; therefore, an objective function and optimisation algorithm [the non-dominated sorting genetic algorithm-III (NSGA-III)] are introduced in this section.

3.1 Non-dominated sorting genetic Algorithm-III (NSGA-III)

The NSGA-III (Deb and Jain 2014) includes five main steps: (1) Population initialisation: Decision variables are randomly generated based on the given upper and lower boundaries. (2) Offspring selection (this is the selection algorithm for the next generation of individual evolution; it is based on a genetic algorithm): according to natural section theory, offspring selection is introduced in the algorithm to determine which is more likely to produce a solution. (3) Generation of new offspring: The parent generation generates new offspring via crossovers and mutations. (4) Non-dominated sorting: The solutions are sorted according to non-dominated relations. (5) Reference-point-based selection mechanism: To select a new group of size N in the next generation, a reference-point-based selection mechanism is introduced in the NSGA-III; this guarantees the uniformity of the distribution and is an enhanced optimisation drive for multi-objective optimisation problems. We expected the entire process to identify each group member corresponding to the reference point close to the Pareto optimal frontier. These group members constitute a set of Pareto-optimal solutions.

3.2 Model proposal

The structure and construction process of the proposed two-objective combined model CM are introduced in this section.
1. We select the data to be tested and use SSA (Abdollahzade et al. 2015) to denoise it.

(1) Embedding:
We construct a trajectory matrix $X_{L \times K}$ as

$$X = [X_1, \ldots, X_N] = (x_{ij})_{i,j=1}^{L,K}$$

$$= \begin{bmatrix}
  x_1 & x_2 & x_3 & \cdots & x_K \\
  x_2 & x_3 & x_4 & \cdots & x_{K+1} \\
  x_3 & x_4 & x_5 & \cdots & x_{K+2} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  x_L & x_{L+1} & x_{L+2} & \cdots & x_N \\
\end{bmatrix} \quad (1)$$
for the time series data, where $X = (x_1, \ldots, x_{L-1})^T$, $(1 < i < K, L \leq K)$ is a lag vector of length $L$, and $X$ is a Hankel matrix for which the elements on each of the sub diagonals are equal.

(2) Singular value decomposition

Singular value decomposition (SVD) is applied to the trace matrix $X$. We let $S = XX^T$ and calculate the eigenvalues $\lambda_1, \ldots, \lambda_L$ of $S$; because $S$ is a symmetric matrix, we have that $\lambda_1 \geq \cdots \geq \lambda_L \geq 0$, and the standard orthogonal basis of the matrix $S$ corresponding to these eigenvalues is obtained as $U_1, \ldots, U_L$.

Let $d = \text{rank}(X)$ and $V_i = X^T U / \sqrt{\lambda_i}, (i = 1, \ldots, d)$; then, the SVD of the trace matrix $X$ can be expressed as $X = X_1 + \cdots + X_N$, $X_i = \sqrt{\lambda_i} U_i V_i^T, (i = 1, \ldots, d)$. The rank of every matrix $X_i$ is one, and these matrices are referred to as elementary. Vectors $U_i$ are left as singular vectors of matrices $X_i$, and set $V_i, (i = 1, \ldots, d)$ is called the spectrum of the trace matrix; hence, this is a singular spectrum decomposition.

(3) Grouping eigenvalues:

By grouping the elementary matrices $X_i (i = 1, \ldots, d)$, the index set $\{1, \ldots, d\}$ is divided into $m$ disjoint subsets, $I_1, \ldots, I_m$. Let $I = \{i_1, \ldots, i_p\}$. Then, the matrices $X_l$ corresponding to Group I are defined as $X_l = X_{i_1} + \cdots + X_{i_p}$. Because the subsets are divided into $m$ groups, $X$ can be expressed as $X = X_1 + \cdots + X_N$.

(4) Diagonal average:

Each matrix $X_i$ of the group decomposition is Hankelised; then, by forming a one-to-one correspondence between the obtained Hankel matrix and time-series data, the previously obtained Hankel matrix will be transformed into a new sequence of length $N$. For any matrix $Z_{L \times K}$, its elements are $z_{ij}$, and Hankel $Z_{L \times K}$; then, the $k$th value denotes the average over all elements of $XZ$ satisfying $i + j = k + 2$.

The diagonal average method is applied to generate a reconstruction sequence $X^{(k)} = (X_1^{(k)}, \ldots, X_N^{(k)})$ from the generated matrix $X_1, \ldots, X_N$. In this way, the initial sequence $x_1, \ldots, x_N$ is decomposed into the sum of $m$ reconstruction subsequences:

$$x_n = \sum_{k=1}^{m} x_n^{(k)}, (n = 1, 2, \ldots, N).$$

$$3. \text{For the N selected single models } M_i (i = 1, 2, \ldots, N), \text{ the forecasting value is set to } \hat{y}_t, \text{ the initial weight } \hat{w}_i \text{ is given (to construct a combined model M_0), and the forecasting value is } \hat{y}_t = \sum_{i=1}^{M} \hat{w}_i \hat{y}_i, t = 1, 2, \ldots, L, \text{ where } t \text{ represents each time point in the time series. In CM, to achieve a high accuracy and stability, the objective functions can be defined as}

$$\text{Minimise} = \left\{ \begin{array}{l} f_1 = \text{MSE} (\tilde{Y}, Y) \\ f_2 = \text{VarSE} (\tilde{Y}, Y) \end{array} \right.,$$

where

$$\text{MSE}(\tilde{Y}, Y) = \frac{1}{L} \sum_{t=1}^{L} \left( \sum_{i=1}^{M} \hat{w}_i \hat{y}_i - y_t \right)^2,$$

$$\text{VarSE}(\tilde{Y}, Y) = \frac{1}{L} \sum_{t=1}^{L} \left( \sum_{i=1}^{M} \hat{w}_i \hat{y}_i - y_t \right)^2 \quad \text{and} \quad -\frac{1}{L} \sum_{t=1}^{L} \left( \sum_{i=1}^{M} \hat{w}_i \hat{y}_i - y_t \right)^2.$$
Datasets 1002 to 1145 were used for testing; thus, we were used for testing, and the first value was forecasted. The training and test sets were kept unchanged. For forecasting the second value, the training set and test set ratio between the training and test sets was 125:18. When differed slightly depending on the forecasting steps. The datasets were selected as the research objects, and the data CM, 10-min wind speed datasets from four stations were generated reference points: 10; maximum number of iterations: 50; population size: 80; crossover percentage: 0.5; mutation percentage: 0.5; and mutation rate: 0.02.

### 4 Numerical experiment

To evaluate the forecasting accuracy and stability of our CM, 10-min wind speed datasets from four stations were selected for multi-step forecasting. In this study, three datasets were selected as the research objects, and the data differed slightly depending on the forecasting steps. The ratio between the training and test sets was 125:18. When forecasting the second value, the training set and test set were each moved one dataset forward, and the numbers of the training and test sets were kept unchanged. For example, when forecasting the first value, the first 1000 sets of data were used for training. Datasets 1001 to 1144 were used for testing, and the first value was forecasted. When the second value was needed for forecasting, the second set (up to Dataset 1001) was used for training, and Datasets 1002 to 1145 were used for testing; thus, we obtained the second forecasting value; the process was repeated for a total of 1008 forecasted values. This method stopped learning when the training error reached \( \text{MSE} = 10^{-6} \) (after normalisation). Three datasets were used for the one-step, two-step, and three-step forecasting.

#### 4.1 Performance metrics

To evaluate the characteristics of the model more comprehensively, certain performance indices were considered. Eight metrics, mean absolute error (MAE), RMSE, standard deviation of absolute percentage error (STDAPE), direction accuracy (DA), Theil U statistic 1 of forecasting results (U1), Theil U statistic 2 of forecasting results (U2), MAPE, and coefficient of determination \( R^2 \), were used, as shown in Table 1; furthermore, the Diebold–Mariano (DM) test and forecasting availability test were used. These metrics were taken from a study by Wang et al. (2018).

### 4.2 Experiment I: One-step-ahead forecasting comparison between our proposed combined model (CM) and other models

This experiment aimed to compare the performance of our forecasting model against some widely used statistical forecasting models, ANNs, and other established models. The forecasting results for one-step-ahead forecasting are shown in Table 2 and Fig. 2. We can see that extreme learning machine (ELM) consistently obtained the optimal results at Sites 1, 2, and 3. At Site 4, the adaptive network-
based fuzzy inference system (ANFIS) performance was optimal among the branch models. CM consistently realised the optimal performance of all models.

Table 2 and Fig. 2 show the numerical simulation results and model performance, respectively.

1. At Site 1, for the branch models, ELM achieved the best results in terms of MAE, RMSE, U1, U2, and \( R^2 \). The STDAPE value (4.96%) of ELM was 0.01% higher than those of ARIMA and ANFIS, which achieved a value of 4.95%. The DA ensures that the trend of the forecasting results is consistent with the true data; for this metric, ANFIS produced an optimal value of 43.20%. ANFIS also achieved the lowest MAE value, of 5.79%.

2. For Site 2 in Table 2 and Fig. 2, the optimal results of MAE, RMSE, STDAPE, U1, and MAPE were obtained by the ELM for the five single models. ARIMA
achieved the best U1 and $R^2$ values, at 0.0327 and 0.9809, respectively. ANFIS optimally ensured that the trend of forecasting results was consistent with the original data (with a value of 46.47% DA).

3. The inequality coefficients (U1 and U2) were also effective for measuring the predictive powers of the models. The ELM obtained better values (0.0316 and 1.3986 for U1 and U2, respectively) for Site 3 than the other models. ELM also performed best in terms of MAE, RMSE, MAPE, and $R^2$. ARIMA and ANFIS achieved the best value of STDAPE and DA, with values of 5.33% and 41.61%, respectively.

Fig. 2 Forecasting results for the one-step-ahead forecast
4. To illustrate the relationship between the true and forecasting values, $R^2$ was employed in this study; furthermore, in the results at Site 4, ELM performed better than the other single models. For the other metrics, ANFIS obtained the best results for all single models.

Remarks

1. No single model could obtain the best results for all metrics in Table 2.
2. Except for Site 4, the ELM performed better than every other single model. The weight of the ELM was highest among the four combined models for the four datasets.
3. For the four different combined models of the four sites, the sub-models and weights were different. For the different datasets, no single model performed best. To achieve the best forecasting results, the sub-models had to be modified by the dataset.
4. For the combined model, five models with the lowest MAPE were selected as the single models in CM (e.g. the BPNN at Site 3, with a value of 6.74%). This is because the MAPE values for the single models were not the only indexes used to select the sub-models and build CM.
5. For the single models selected to build CM, each achieved one or more optimal scores according to the metrics.

4.3 Experiment II: Two-step-ahead forecasting comparison between our proposed model (CM) and other models

This experiment aimed to compare the two-step-ahead forecasting performance of CM with some widely used statistical forecasting models, ANNs, and other established models. Table 3 and Fig. 3 show the two-step-ahead forecasting results for the four sites. From these results, ELM performed better than the other single models at the four sites (four datasets), and CM obtained the optimal values of all models under the eight metrics.

Table 3 and Fig. 3 show the results, which are as follows:

1. For Site 1, LSTM achieved the optimal results for the MAE, DA, and MAPE, with values of 0.3257, 58.59%, and 4.81%, respectively. The other optimal values of the metrics were obtained by ELM. The weight of ELM was the highest among the single models, at 0.5689. The other models selected for CM were ARIMA, ANFIS, GRNN, and LSTM, with weights of 0.0199, 0.1149, 0.0568, and 0.2404, respectively.
2. ELMAN optimally ensured that the trend of the forecasting result was consistent with the true data, with a DA value of 55.61% at Site 2. ELM achieved the best values under the other metrics. The combined model was built using the ELM, ARIMA, ANFIS, SVM, and BPNN. ELMAN was not selected to build CM, although the forecasting trend of ELMAN more resembled the original data than the other single models.
3. In the forecasting results for the two-step-ahead forecast (shown in Table 3 and Fig. 3), ELM performed best among the nine single models in seven metrics, except for DA. However, the weight of ELM did not exceed 0.5 (it was 0.4914). The trend of the results forecasted by BPNN most resembled that of the original data.
4. The MAE value can better reflect the actual forecasting value error; here, the BPNN obtained the lowest MAE value (0.3197). The BPNN also obtained the best result for DA, with a value of 61.67%. For the data of Site 4, ELM achieved the best values of RMSE, STDAPE, U1, U2, MAPE, and $R^2$, with values of 0.4633, 4.13%, 0.0306, 1.2056, 4.71%, and 0.9813, respectively.

Remarks

1. For the two-step-ahead forecasting (see Table 3), the optimal results in all metrics could not be obtained from any single model.
2. The forecasting results of ELM almost achieved the optimal values at the four sites, as shown in Table 3 and Fig. 3. The ELM weights selected to build CM for the four sites were 0.5698, 0.5249, 0.4914, and 0.5994, respectively.
3. At Site 2, ELMAN achieved the best DA value; however, the CM for Site 2 was not built by ELMAN. This indicates that a single standard or metric cannot determine whether a single model should be selected to construct CM.
4. For Sites 3 and 4, the models selected to build CM were the same. The single models were ELM, ARIMA, ANFIS, GRNN, and BPNN. However, the weights of these models for Sites 3 and 4 were different. This indicates that the CMs for these two sites differed.
5. For the four sites or four datasets, the CMs differed.
6. ELMAN and BPNN were not selected to construct the CMs for the four sites.
4.4 Experiment III: Three-step-ahead forecasting comparison between our proposed model (CM) and other models

This experiment aimed to compare the three-step-ahead forecasting performance of CM with several widely used statistical forecasting models, ANNs, and other established models. ELM and BPNN outperformed the other single models at the four sites (four datasets).

Table 3  Two-step-ahead forecasting performance of four sites (the optimal results of single models are highlighted in bold)

| Metric      | ELM    | ARIMA  | ANFIS   | GRNN    | SVM     | BPNN    | ELMAN   | RBFNN   | LSTM    | CM     |
|-------------|--------|--------|---------|---------|---------|---------|---------|---------|---------|--------|
| Site 1      | MAE    | 0.3554 | 0.4193  | 0.4065  | 0.4261  | 0.5541  | 0.5862  | 0.5939  | 0.5333  | 0.3257  | 0.3184 |
|             | RMSE   | 0.5046 | 0.5519  | 0.5605  | 0.7745  | 0.7654  | 0.8129  | 0.7953  | 0.7069  | 0.7628  | 0.4610 |
|             | STDAPE| 4.50%  | 5.37%   | 5.10%   | 7.28%   | 7.28%   | 7.64%   | 7.65%   | 6.60%   | 11.39%  | 4.60%  |
|             | DA     | 48.96% | 35.05%  | 45.98%  | 45.18%  | 31.08%  | 29.31%  | 28.92%  | 26.32%  | 58.59%  | 55.51% |
|             | U1     | 0.0336 | 0.0367  | 0.0371  | 0.0512  | 0.0508  | 0.0512  | 0.0539  | 0.0469  | 0.0505  | 0.0506 |
|             | U2     | 1.2511 | 1.3907  | 1.4880  | 1.9633  | 2.2347  | 2.3641  | 2.4449  | 1.9858  | 2.1216  | 1.0712 |
|             | MAPE   | 5.03%  | 6.20%   | 5.74%   | 6.04%   | 7.98%   | 8.32%   | 8.02%   | 7.75%   | 4.81%   | 4.53%  |
|             | \(R^2\) | 0.9775 | 0.9729  | 0.9737  | 0.9511  | 0.9497  | 0.9116  | 0.9401  | 0.9565  | 0.9487  | 0.9812 |
|             | Weight | 0.5689 | 0.0199  | 0.1149  | 0.0568  | –       | –       | –       | –       | 0.2404  | 1      |

CM: Proposed combined model

4.4 Experiment III: Three-step-ahead forecasting comparison between our proposed model (CM) and other models

This experiment aimed to compare the three-step-ahead forecasting performance of CM with several widely used statistical forecasting models, ANNs, and other established models. ELM and BPNN outperformed the other single models at the four sites (four datasets).

Table 4 and Fig. 4 show the results, which can be summarised as follows:

1. From the results of Site 1, the proportion of variation in the dependent variable of indicator \(R^2\) can be explained by the independent variable via a regression relationship. LSTM achieved the optimal value (0.9709). The optimal values of U1 and RMSE were also achieved by LSTM, at 0.6707 and 32.96%, respectively. The RMSE
measures the deviation between the observed and real data. It is often used as a standard to measure the forecasting results of machine-learning models. ELM achieved the optimal values under the other five metrics: MAE, STDAPE, U1, U2, and MAPE.

2. For the experiment at Site 2, ELM and SVM were the two best of the nine single models, and they obtained the best values in five and three metrics, respectively. Although SVM only achieved the three best values of
the metrics (lower than ELM), the weight of the SVM used to build CM was 0.5055, higher than that of ELM.

3. At Site 3, the BPNN performed best among the nine single models, and the weight of the BPNN used to build CM was 0.4829, which was the highest among the nine single models.

4. Similar to Site 3, an optimal single model was observed at Site 4; ELM had an overwhelming advantage among the nine models. The CM of Site 4 was built using ELM, ARIMA, ANFIS, GRNN, and LSTM.

Remarks:

1. For the three-step-ahead forecasting from Table 4, the best results of all metrics in four sites could not be obtained by any single model.
Fig. 4  Forecasting results of three-step-ahead forecast
2. ELM, ARIMA, and ANFIS were selected to build the three-step-ahead forecasting CM for the four sites, which indicated that these four models were fit for three-step-ahead forecasting and the four datasets.

3. For the four sites, the CMs were built with different single models, to obtain the best CM for the forecasting results.

4. The weight of LSTM was 0.6189, greater than the weight of ELM; however, ELM obtained the best metric scores. Thus, the weights of the branch models were unrelated.

5. For Sites 1 and 4, the single models of the combined model were identical. The single models were ELM, ARIMA, ANFIS, GRNN, and LSTM. However, the weights of these models for Sites 1 and 4 differed. This indicates that the CMs for these two sites were different. Therefore, for the four sites or four datasets, the CMs chosen by our proposed method differed.

6. ELMAN and RBFNN were not selected to build CMs for the four sites.

7. For constructing CM, five models with the lowest MAPE were selected, including the ELMAN at Site 4 (with a value of 6.59%). This is because the MAPE of the single models did not propose a single index to help select the branch model of the CM.

4.5 Experiment IV: DM test and forecasting availability

To further evaluate our CM, we used two evaluation methods—the DM test and forecasting availability—to evaluate the model quality.

| Table 5 Results for the Diebold and Mariano (DM) test |
|------------------------------------------------------|
| Site | Site 1 | Site 2 | Site 3 | Site 4 |
| One-Step |
| ELM versus CM | 1.3065* | 0.1395* | 0.1661* | 1.4595* |
| ARIMA versus CM | 1.6757* | 1.6586* | 1.6739* | 1.8051* |
| ANFIS versus CM | 0.0274* | 1.3118* | 1.4458* | 0.6710* |
| GRNN versus CM | 1.5687* | 1.7102* | 1.7210* | 1.6786* |
| SVM versus CM | 2.4982** | 1.9551* | 2.6040** | 2.2348** |
| BPNN versus CM | 1.8466* | 2.5931*** | 1.9504* | 2.0062** |
| ELMAN versus CM | 6.7752*** | 6.8146*** | 6.8772*** | 7.0040*** |
| RBFNN versus CM | 2.0196** | 2.1254* | 2.1854* | 2.6704*** |
| LSTM versus CM | 3.3721*** | 4.5251*** | 4.8914*** | 6.0206*** |
| Two-Step |
| ELM versus CM | 1.2993* | 0.1374* | 0.1637* | 0.6674* |
| ARIMA versus CM | 1.8668* | 1.6942* | 1.6809* | 1.9734** |
| ANFIS versus CM | 1.5772* | 1.6655* | 1.7305* | 1.6777* |
| GRNN versus CM | 1.6942* | 2.0944** | 1.9513* | 1.8093* |
| SVM versus CM | 2.4858* | 6.3136*** | 2.5871*** | 5.7858*** |
| BPNN versus CM | 6.2349*** | 1.9270* | 1.4301* | 1.4517* |
| ELMAN versus CM | 3.2725*** | 1.3073* | 2.1733** | 2.2234*** |
| RBFNN versus CM | 2.0444** | 2.5292* | 4.6024*** | 2.6808*** |
| LSTM versus CM | 0.0271* | 4.3469*** | 6.3290*** | 6.3992*** |
| Three-Step |
| ELM versus CM | 0.0282* | 1.3041* | 1.4711* | 0.6785* |
| ARIMA versus CM | 1.7416* | 1.6725* | 1.7229* | 1.8285* |
| ANFIS versus CM | 1.6226* | 1.9251* | 1.7691* | 1.7090* |
| GRNN versus CM | 1.9223* | 2.4704** | 1.9977** | 2.0000** |
| SVM versus CM | 6.5018*** | 0.1377* | 6.7895** | 6.3215*** |
| BPNN versus CM | 2.5322** | 1.7014* | 0.1693* | 2.6339*** |
| ELMAN versus CM | 3.3788*** | 2.0647** | 2.6412*** | 2.2112** |
| RBFNN versus CM | 2.0801** | 4.2901** | 4.6880*** | 5.7065*** |
| LSTM versus CM | 1.3544* | 6.56452*** | 2.1912** | 1.4819* |

*1% significant difference; **5% significant difference; ***10% significant difference; CM: our proposed combined model
DM testing, proposed by Diebold and Mariano (1995), focuses on forecasting accuracy and evaluates the forecasting performance of two or more time-series models, as well as forecasting availability (Zhang et al. 2018). The effectiveness of the forecasting was measured by the sum of the squares and the mean square deviation of the forecasting errors, to further evaluate and analyse the performance of CM. Among all models, the optimal performance was achieved by CM.

The results of the DM test and forecasting availability are shown in Tables 5 and 6, respectively.

(1) As shown in Table 5, CM differed considerably from the other models, regardless of the dataset or order.

(2) The results of forecasting availability are listed in Table 6. In wind speed forecasting, the first- and second-order forecasting availability of CM for the four datasets and one-step, two-step, and three-step forecasting outperformed those of the other models. For example, at Site 1 (one-step forecasting), the first-order forecasting availabilities of each model were 0.9419, 0.9400, 0.9421, 0.9401, 0.9098, 0.9383, 0.8582, 0.9222, 0.8945, and 0.9504, respectively.

**Table 6** Forecasting availability results (the optimal results of single models are highlighted in bold)

| Site | Forecasting availability | Forecasting step | ELM | ARIMA | ANFIS | GRNN | SVM | BPNN | ELMAN | RBFNN | LSTM | CM |
|------|--------------------------|------------------|-----|-------|-------|------|-----|------|-------|-------|------|-----|
| 1-Order |
| One-Step | 0.9419 | 0.9400 | 0.9421 | 0.9401 | 0.9098 | 0.9383 | 0.8582 | 0.9222 | 0.8945 | 0.9504 |
| Two-Step | 0.9497 | 0.9380 | 0.9426 | 0.9396 | 0.9202 | 0.9383 | 0.8582 | 0.9222 | 0.8945 | 0.9504 |
| Three-Step | 0.9147 | 0.9121 | 0.9159 | 0.9126 | 0.8908 | 0.9098 | 0.8797 | 0.8843 | 0.8880 | 0.9293 |
| Two-Step | 0.9259 | 0.9093 | 0.9158 | 0.9112 | 0.8847 | 0.8797 | 0.8843 | 0.8880 | 0.8880 | 0.9336 |
| Three-Step | 0.8970 | 0.8881 | 0.8917 | 0.8843 | 0.8437 | 0.8665 | 0.8559 | 0.8711 | 0.8921 | 0.9314 |
| 2-Order |
| One-Step | 0.9464 | 0.9451 | 0.9458 | 0.9378 | 0.9319 | 0.9202 | 0.9470 | 0.9223 | 0.8964 | 0.9555 |
| Two-Step | 0.9550 | 0.9442 | 0.9446 | 0.9320 | 0.9166 | 0.9411 | 0.9491 | 0.9230 | 0.9204 | 0.9568 |
| Three-Step | 0.9402 | 0.9310 | 0.9298 | 0.9217 | 0.9447 | 0.9305 | 0.9268 | 0.9125 | 0.8751 | 0.9481 |

The higher value, the better the forecasting effectiveness of the model CM: Our proposed combined model

DM testing, proposed by Diebold and Mariano (1995), focuses on forecasting accuracy and evaluates the forecasting performance of two or more time-series models, as well as forecasting availability (Zhang et al. 2018). The effectiveness of the forecasting was measured by the sum of the squares and the mean square deviation of the forecasting errors, to further evaluate and analyse the performance of CM. Among all models, the optimal performance was achieved by CM.

The results of the DM test and forecasting availability are shown in Tables 5 and 6, respectively.

(1) As shown in Table 5, CM differed considerably from the other models, regardless of the dataset or order.

(2) The results of forecasting availability are listed in Table 6. In wind speed forecasting, the first- and second-order forecasting availability of CM for the four datasets and one-step, two-step, and three-step forecasting outperformed those of the other models. For example, at Site 1 (one-step forecasting), the first-order forecasting availabilities of each model were 0.9419, 0.9400, 0.9421, 0.9401, 0.9098, 0.9383, 0.8582, 0.9222, 0.8945, and 0.9504, respectively.

**Remark**

(1) The results of the DM test showed that the CM differed from other models. The higher the value, the greater this difference.

(2) The forecasting availability results show the differences between CM and other models. The higher the value, the greater the difference. The results show that the first- and second-order values of the CMs are close to 1, which indicates that CM is significantly better than the other models.
5 Summary

From the four experiments, we obtained the following findings:

1. The sub-models of CM were not static. Across the three different datasets, four sites, and different step-ahead-forecasting scenarios, the preferred model was always varied to achieve the best results.

2. No single model could consistently achieve the best results among the sub-models, owing to the complexity of the data.

3. The multi-objective optimisation algorithm was used to optimise the weights of the combined model’s sub-models from the former three experiments. The weights of the sub-models in CM differed. In addition, the multi-objective optimisation algorithm could balance the objective functions of the combined model, thereby ensuring the accuracy and effectiveness of the forecasting.

4. The model selection chose the best sub-models to construct CM in different situations (datasets, forecasting steps, and sites); this made the selection of the sub-models more reasonable, and the combined model helped achieve the best results.

Our experiments demonstrate that CM has a stronger forecasting power and higher forecasting accuracy than the benchmark model.

6 Conclusion

In this study, our proposed combined model obtained an optimal result compared to the other models. There was no single ANN that could perfectly solve the problem for different levels of step-ahead forecasting and different datasets. Furthermore, based on the experiments, certain innovations of the forecast system developed in this study [i.e. our proposed model (CM)] considered not only different time and site data but also the disadvantages of the combined model. To overcome these disadvantages, a model was proposed using multiple objective functions, and the model selection theory and innovations can be summarised as follows:

1. The multi-objective functions guaranteed the stability and accuracy of the CM’s results, because they considered both aspects. (2) The model selection theory made the single models of the CM more reasonable, instead of manually selecting single models. (3) Our proposed model confirmed that for different data and different step-ahead forecasting processes, the optimal combined model was not fixed. For these results, it was necessary to adjust the sub-models used to construct CM for the different data. These developments in our model are rarely seen in other studies; therefore, this study fills that research gap.

To summarise, by overcoming the disadvantages and making innovations, our proposed model based on multiple objective functions and model selection was found to be stable and accurate (the MAPE is less than 4.60% and the STDAPE is less than 5.68%); it overcame the difficulty of selecting sub-models for the combined model. With the forecasting results and theories, CM can also be applied to futures, forwards, securities, house prices, and other forecasting fields. More benchmark models should be added to the model selection, to help CM achieve better results. However, more models will increase the model runtime; thus, the number of models and runtime should be kept balanced in future studies.

Appendix

For the data, only with a better understanding of the features of the data we can better select the model to prepare for future work. In order to achieve better results, we must consider the characteristics of the data. Generally speaking, the linear model has a better fitting effect for linear data, as the nonlinear model does for nonlinear data.

Only when we understand the characteristics of the data we can achieve good results in future forecasting work. For the data, it is not just linear or nonlinear, but both. Therefore, it is necessary to judge the linear nonlinearity of the data used in this paper, so we constructed the above experiments.

From the results of Tables 7 and 8, wind speed data are both linear and nonlinear by hypothesis test. So, the linear models and nonlinear models considered in our proposed forecasting model are correct and necessary.

Basic methods and theories

Support vector machines

The support vector machine (SVM) method is a machine learning method that was proposed by Vapnik (1997) and the nonlinear function can be written as follows:

\[ f(x) = \omega^T \phi(x) + b \]  

where \( \phi(x) \) is the kernel function that maps the data from low-dimensional space to high dimensional space. \( \omega \) and \( b \) are the coefficient and threshold, respectively.

Using loss function \( \varepsilon \) to optimize regression function, and the best regression function is found by the minimum value of the loss function. \( \varepsilon \) is as follows:
Wind speed forecast based on combined theory, multi-objective optimisation, and sub-model selection

Table 7 Testing wind speed data by adjusting to linear functions or nonlinear functions

| Function | NO | EDF | $R^2$ | AR | RMSE | $p$ value |
|----------|----|-----|-------|----|------|-----------|
| Site1    | $f_1(x)$ | 1008 | 1001 | 0.9235 | 0.9230 | 24.3735 | 0 |
| $f_2(x)$ | 1008 | 1001 | 0.9184 | 0.9179 | 25.1786 | 0 |
| $f_3(x)$ | 1008 | 1001 | 0.9089 | 0.9083 | 26.5994 | 0 |
| $f_4(x)$ | 1008 | 1001 | 0.9193 | 0.9188 | 25.0372 | 0 |
| Site2    | $f_1(x)$ | 1008 | 1001 | 0.9301 | 0.9297 | 22.7501 | 0 |
| $f_2(x)$ | 1008 | 1001 | 0.9233 | 0.9229 | 23.8312 | 0 |
| $f_3(x)$ | 1008 | 1001 | 0.9194 | 0.9189 | 24.4343 | 0 |
| $f_4(x)$ | 1008 | 1001 | 0.9275 | 0.9271 | 23.1694 | 0 |
| Site3    | $f_1(x)$ | 1008 | 1001 | 0.9322 | 0.9318 | 23.6373 | 0 |
| $f_2(x)$ | 1008 | 1001 | 0.9230 | 0.9225 | 25.1963 | 0 |
| $f_3(x)$ | 1008 | 1001 | 0.9256 | 0.9252 | 24.7628 | 0 |
| $f_4(x)$ | 1008 | 1001 | 0.9309 | 0.9305 | 23.8684 | 0 |
| Site4    | $f_1(x)$ | 1008 | 1001 | 0.9260 | 0.9255 | 22.4048 | 0 |
| $f_2(x)$ | 1008 | 1001 | 0.9188 | 0.9184 | 23.4604 | 0 |
| $f_3(x)$ | 1008 | 1001 | 0.9126 | 0.9121 | 24.3490 | 0 |
| $f_4(x)$ | 1008 | 1001 | 0.9226 | 0.9221 | 22.9103 | 0 |

| Function | NO | EDF | $R^2$ | AR | RMSE | $p$ value |
|----------|----|-----|-------|----|------|-----------|
| Site1    | $f_1(x)$ | 1008 | 1001 | 0.8586 | 0.8578 | 20.1457 | 0 |
| $f_2(x)$ | 1008 | 1001 | 0.8532 | 0.8523 | 20.5303 | 0 |
| $f_3(x)$ | 1008 | 1001 | 0.8534 | 0.8525 | 20.5149 | 0 |
| $f_4(x)$ | 1008 | 1001 | 0.8575 | 0.8567 | 20.2265 | 0 |
| Site2    | $f_1(x)$ | 1008 | 1001 | 0.8440 | 0.8431 | 21.1436 | 0 |
| $f_2(x)$ | 1008 | 1001 | 0.8160 | 0.8149 | 22.9623 | 0 |
| $f_3(x)$ | 1008 | 1001 | 0.8444 | 0.8434 | 21.1190 | 0 |
| $f_4(x)$ | 1008 | 1001 | 0.8442 | 0.8433 | 21.1284 | 0 |
| Site3    | $f_1(x)$ | 1008 | 1001 | 0.8594 | 0.8586 | 20.3246 | 0 |
| $f_2(x)$ | 1008 | 1001 | 0.8546 | 0.8537 | 20.6712 | 0 |
| $f_3(x)$ | 1008 | 1001 | 0.8535 | 0.8527 | 20.7459 | 0 |
| $f_4(x)$ | 1008 | 1001 | 0.8584 | 0.8576 | 20.3987 | 0 |
| Site4    | $f_1(x)$ | 1008 | 1001 | 0.8366 | 0.8356 | 21.8374 | 0 |
| $f_2(x)$ | 1008 | 1001 | 0.8130 | 0.8118 | 23.3635 | 0 |
| $f_3(x)$ | 1008 | 1001 | 0.8368 | 0.8358 | 21.8239 | 0 |
| $f_4(x)$ | 1008 | 1001 | 0.8367 | 0.8357 | 21.8293 | 0 |

Constraints:

\[ \varepsilon = \min \frac{1}{2} \omega^T \omega + C \sum_{i=1}^n (\xi_i + \hat{\xi}_i) \]  

(9) \[
\begin{align*}
\left\{
\begin{array}{l}
y_i - [\omega, \phi(x_i)] - b \leq \xi_i + \varepsilon \\
y_i + [\omega, \phi(x_i)] + b - y_i \leq \xi_i + \varepsilon \\
\xi_i \geq 0, \hat{\xi}_i \geq 0
\end{array}
\right.
\]

where \( C \) is a penalty factor; \( \xi_i, \hat{\xi}_i \) are slack variables, \( b \) is the offset, and \( x_i, y_i \) are the input and output, respectively.
Table 8 The explanations of the test parameters

| Parameter                                | Description                                                                 |
|------------------------------------------|-----------------------------------------------------------------------------|
| Number of observations                   | Number of rows without any NaN values                                       |
| Error degrees of freedom                 | \( n - p \), where \( n \) is the number of observations, and \( p \) is the number of coefficients in the model, including the intercept |
| \( R^2 \) and Adjusted \( R^2 \)-squared  | Coefficient of determination and adjusted coefficient of determination, respectively |
| F-statistic versus constant model        | Test statistic for the F-test on the regression model. It tests for a significant regression relationship between the response variable and the predictor variables |
| \( p \) value                            | \( p \) value for the F statistic of the hypotheses test that the corresponding coefficient is equal to zero or not |

After the operation, the linear regression function can be obtained as follows:

\[
f(x) = \sum_{i=1}^{n}(\beta_i - x_i)K(x,y_i) + b
\]  

(11)

where \( K(x,y_i) \) is the kernel function of SVM.

**Long short-term memory network**

The long short-term memory (LSTM) network is a recurrent neural network (RNN), which was proposed by Hochreiter and Schmidhuber (Hochreiter and Schmidhuber 1997) in 1997 and can be written as follows (Greff et al. 2017):

\[
z^t = g(\omega_z x^t + r_z y^{t-1} + b_z)
\]

(12)

\[
i^t = \sigma(\omega_i x^t + r_i y^{t-1} + p_i c^{t-1} + b_i)
\]

(13)

\[
f^t = \sigma(\omega_f x^t + r_f y^{t-1} + p_f c^{t-1} + b_f)
\]

(14)

\[
c^t = i^t f^t + c^{t-1} \neg f^t
\]

(15)

\[
o^t = \sigma(\omega_o x^t + r_o y^{t-1} + p_o c^t + b_o)
\]

(16)

\[
y^t = h(c^t) o^t
\]

(17)

\[
\sigma(x) = \frac{1}{1 + e^{-x}}
\]

(18)

\[
g(x) = h(x) = \tanh(x)
\]

(19)

where \( t \) is the time; \( N \) is the cells of LSTM; \( \omega_z, \omega_i, \omega_f, \omega_o \in R^{N \times M} \) are the input weights; \( r_z, r_i, r_f, r_o \in R^{N \times M} \) are the recurrent weights; \( p_i, p_f, p_o \in R^N \) are the peephole weights (Gers and Schmidhuber 2000); \( b_z, b_i, b_f, b_o \in R^N \) are the biases; \( g(x), h(x) \) and \( \sigma(x) \) are activation functions; \( z^t \) is the activation of the input block; \( i^t \) is the activation of the forget gate; \( c^t \) is the cell state at time \( t \); \( o^t \) is the activation of the output gate; and \( y^t \) is the output of the cell at time \( t \).

**Autoregressive integrated moving average**

The autoregressive integrated moving average (ARIMA) model is one of the most popular forecasting models in the wind speed forecasting field (Contreras et al. 2003) and can be written as follows:

\[
\phi(B)(1 - B)^d X_t = \theta(B) \epsilon_t
\]

(20)

\[
\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p
\]

(21)

\[
\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q
\]

(22)

These above formulas are recorded as ARIMA \((p, d, q)\), where \( B^q X_t = X_{t-q} \), \( X_t \) is a time series at time \( t \), \( \epsilon_t \) is the random error at time \( t \), and \( B \) is the backward shift operator.

**Back-propagation neural network**

Back-propagation neural network (BPNN) is a widely used multi-layer feedforward neural network that is based on a gradient descent method that minimizes the sum of the squared errors between the actual output value and the expected output value. The output function is between 0 and 1, which can convert input to output to achieve continuous non-linear mapping (Guo et al. 2011) and can be written as follows:

The topology of the BPNN is as follows:

\[
X' = \{X'_i\} = 2 \frac{X_i - X_{i_{\min}}}{X_{i_{\max}} - X_{i_{\min}}} - 1, (i = 1, 2, \ldots, n), X' \in [-1, 1]
\]

(23)

where \( X_{i_{\min}} \) and \( X_{i_{\max}} \) are the minimum and maximum value of the input array or output vectors, and \( X'_i \) denotes the real value of each vector.

Step 1. Calculate the outputs of all hidden layer nodes:

\[
y_j = f(\sum_i w_{ij} x_i + b_j) = f(\text{net}_j)(j = 1, \ldots, 2n)
\]

(24)
where the activation value of node $j$ is $net_j$, $w_{ij}$ represents the connection weight from input node $i$ to hidden node $j$, $b_j$ represents the bias of neuron $j$, $y_j$ represents the output of hidden layer node $j$, and $f$ is the activation function of a node, which is usually a sigmoid function.

Step 2. Calculate the output data of the neural network:

$$O_1 = f_0 \left( \sum_j w_{0j} y_i + b_0 \right), \quad (i = 1, \ldots, 2n)$$

(26)

where $w_{0j}$ represents the connection threshold from hidden node $j$ to the output node, $b_0$ represents the bias of the neuron, $O_1$ represents the output data of the network, and $f_0$ is the activation function of the output layer node.

Step 3. Minimize the global error via the training algorithm:

$$\text{Mean Square Error} = \frac{1}{m} \sum \left( O_1 - Z \right)^2$$

(27)

where $Z$ represents the real data vector of the output, and $m$ represents the number of outputs.

**Generalized regression neural network**

Generalized regression neural networks (GRNN) were proposed by Specht (1991), and the theoretical basis was nonlinear kernel regression analysis. The network was based on nonlinear regression theory and consisted of four layers of neurons: the input layer, pattern layer, summation layer, and output layer.

**Definition 1** Setting the joint probability density function of the random variable $x$ and $y$ to $f(x, y)$, the observed value of $x$ is $X$. Thus, the estimation value as follows:

$$\hat{Y} = E[y|X] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x f(X, y)dy$$

(28)

**Definition 2** To set the probability density function $f(x, y)$, which is unknown, but can be obtained using a nonparametric estimation, we use the sample observation values of $x$ and $y$:

$$f(x, y) = \frac{1}{n \delta^{d+1} (2\pi)^{d+1}/2} \sum_{i=1}^{n} \exp \left( -\frac{(X - X_i)^T (X - X_i)}{2\delta^2} \right) \exp \left( -\frac{(Y - Y_i)}{2\delta^2} \right)$$

(29)

where $X_i$ and $Y_i$ are the sample observation values of $x$ and $y$, respectively; $\delta$ is the smoothing parameter; $n$ is the number of samples; and $p$ is the dimensionality of the random vector $x$. Here, we can calculate $Y$ using $f(x, y)$ instead of $f(x, y)$ in formula (30). Finally,

$$\hat{Y}(X) = \sum_{i=1}^{n} Y_i \exp \left[ -\frac{(X - X_i)^T (X - X_i)}{2\delta^2} \right]$$

(30)

where $\hat{Y}(X)$ is the weighted average of all sample observations of $Y_i$, and every weight factor $Y_i$ is the Euclidean squared distance index value between the corresponding samples $X_i$ and $X$.

**Radial basis function neural network**

The radial basis function neural network (RBFNN) (Schwenker et al. 2001) is an efficient feedforward neural network that exhibits better approximation performance and global optimal ability than other feedforward networks. The structure of the neural network is simple and the training speed is fast. An RBFNN also includes three layers: an input layer, a hidden layer with a nonlinear activation function (the radial basis function), and an output layer.

The RBFNN was proposed by Schwenker, and this network exhibits better approximation performance and global optimal ability than other feedforward networks. The neural network also has a simple structure and fast training speed. There are three layers: an input layer, an output layer, and a hidden layer with a nonlinear activation function.

**Definition 1** The modelled input is a real-number vector $x \in \mathbb{R}^d$. Then, the output vector of the neural network is a scalar function of the input vector, $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$, given by

$$\varphi(x) = \sum_{i=1}^{N} a_i \rho(\|x - c_i\|)$$

(31)

where $N$ represents the total number of neurons in the hidden layer, the centre vector of the $i$th neuron is represented by $c_i$, and $a_i$ is the weight of neuron $i$ in the linear output neuron.

**Definition 2** A radial basis function is a scalar function that is radially symmetric. Thus, it is defined as a monotone function of the Euclidean distance between any point $x$ in space and $c_i$ of a centre vector. The most commonly used radial basis function is the Gauss
kernel function, given as follows:

\[ k(x - c_i) = \exp\left(\frac{\|x - c_i\|^2}{(2\sigma)^2}\right) \]  

(32)

where \(c_i\) is a centre vector and \(\sigma\) is a width parameter that controls the radial scope of the function.

### Adaptive network-based fuzzy inference system

Jang (1993) combined the best features of the fuzzy system and neural network to construct an adaptive network-based fuzzy inference system (ANFIS). ANFIS integrates the human inference style of fuzzy inference system (FIS) by using input–output sets and a set of if–then fuzzy rules. FIS (Neshat et al. 2012) has structured knowledge, in which each fuzzy rule describes the current behaviour of the system; however, it lacks adaptability to changes in the external environment. Therefore, the concept of neural network learning and FIS are combined in ANFIS (Iñan et al. 2007). ANFIS is a method that uses neural network learning and fuzzy inference to simulate complex nonlinear systems.

Layer 1: This layer is the input layer, which is responsible for the fuzziness of the input signal. Each node I is a node function represented by square node:

\[ O^l_i = \mu_A(x), i = 1, 2 \quad \text{or} \quad O^l_i = \mu_B(y), i = 1, 2 \]  

(33)

where \(x\) (or \(y\)) is the input of node \(i\), \(A_i, B_i\) are fuzzy sets, and \(O^l_i\) is the membership function value of \(A_i\) and \(B_i\), indicating the degree to which \(X\) and \(Y\) belong to \(A_i\) and \(B_i\). Usually, the \(\mu_A\) and \(\mu_B\) are chosen as bell-shaped functions or Gaussian functions. The membership function has some parameters; these parameters are called premise parameters.

Layer 2: The nodes in this layer are responsible for multiplying the input signals and calculating the firing strength of each rule. The output is

\[ O^2_i = \omega_i = \mu_A(x) \times \mu_B(y), i = 1, 2 \]  

(34)

where the output of each node represents the credibility of the rule.

Layer 3: This layer normalizes all applicability, each node is represented by \(N\). The ratio of the \(i\)th rule’s firing strengths to the sum of all rules’ firing strengths is calculated by the \(i\)th node:

\[ O^3_i = \bar{\omega}_i = \omega_i/(\omega_1 + \omega_2)i = 1, 2. \]  

(35)

Layer 4: Calculating the output of the fuzzy rule, the output is

\[ O^4_i = \bar{\omega}_i f_i = (p-x + q-y + r_i) i = 1, 2 \]  

(36)

where \(\bar{\omega}_i\) is the output of layer 3 and \(\{p_i, q_i, r_i\}\) is the parameter set (consequent parameters).

Layer 5: The single node of this layer is a fixed node that calculates the total output of all input signals:

\[ O^5_i = \sum_i \bar{\omega}_i f_i. \]  

(37)

### Extreme learning machine

Extreme learning machine (ELM) is a machine learning algorithm proposed by Huang, which was designed for single-layer feedforward neural networks (SLFNNs) (Huang et al. 2006). The main feature of ELM is that the parameters of hidden layer nodes can be randomly generated without adjustment. The learning process only needs to calculate the output weights.

**Definition 3** An SLFNN consists of three parts: an input layer, hidden layer, and output layer. The output function of the hidden layer is given as follows:

\[ f_L = \sum_{i=1}^n \beta_i h_i(x) = h(x)\beta, \]  

(38)

where \(x\) is the input vector, \(\beta\) is the output weight of the \(i\)th hidden node, and \(h(x)\) is the hidden layer output mapping, called the activation function, defined as follows:

\[ h(x) = G(a_i, b_i, x). \]  

(39)

Here, \(b_i\) is the parameter of the feature mapping (also called the node parameter), and \(a_i\) is called the input weight. In the calculation, the parameter of the feature mapping is randomly initialized and is not adjusted. Hence, the feature mapping of the ELM is also random.

### Elman neural network

The Elman neural network is a machine learning algorithm proposed by Elman designed for SLFNN. In addition to the input layer, the hidden layer, and the output layer, it also has a special contact unit. The contact unit is used to memorize the previous output value of the hidden layer unit. It can be considered as a delay operator. Therefore, the feedforward link part can be corrected for the connection weight, whilst the recursive part is fixed, that is, the
learning correction cannot be performed. The mathematic model of Elman can be written as follows (Elman 1990):

\[ x(k) = f(W^{I1}x(k) + (W^{I2}u(k - 1))) \] (40)
\[ x_i(k) = z_{xi}(k - 1) + x(k - 1) \] (41)
\[ y(k) = W^{I3}x(k) \] (42)

where \( f(x) \) is a sigmoid function, and \( 0 \leq a < 1 \) is a self-connected feedback gain operator. If the \( a = 0 \), then the network is a standard Elman neural network, if the \( a \neq 0 \), then the network is a modified Elman neural network, \( u \) is the input data with \( n \)-dimensional vector, \( x \) is a hidden layer output, \( X_i \) is the hidden layer output with an \( n \)-dimensional vector, \( y \) is the output for the network with an \( m \)-dimensional vector, and \( W^{I1} \), \( W^{I2} \), and \( W^{I3} \), are connection weights with \( n \times n \), \( n \times q \), and \( m \times m \)-dimensional matrices, respectively.

We set the actual output of the k-step system to be \( y_d(k) \), define the error function as \( E(k) = \frac{1}{2}(y_d(k) - y(k))^T(y_d(k) - y(k)) \), and let derivative \( E \) be the connection weights \( A, W^{I1}, W^{I2}, W^{I3} \). The learning algorithm of an Elman network can be obtained using the gradient descent method:

\[ \Delta w^{I3}_{ij} = \eta_3 \delta^0_{i} x_j(k), i = 1, 2, \ldots, m; j = 1, 2, \ldots, n \] (43)
\[ \Delta w^{I2}_{jk} = \eta_2 \delta^b_j u_q(k - 1), j = 1, 2, \ldots, n; q = 1, 2, \ldots, r \] (44)
\[ \Delta w^{I1}_{jl} = \eta_1 \sum_{i=1}^{m}(\delta^0_{i} w^{I3}_{ij} \frac{\partial x_j(k)}{\partial w^{I1}_{jl}}), j = 1, 2, \ldots, n; l = 1, 2, \ldots, n \] (45)

where \( r \) is the node number of the input layer, \( n \) is the node number of the hidden layer and unit layer, and \( m \) is the node number of the output layer. \( \eta_1 \), \( \eta_2 \), and \( \eta_3 \) are the learning steps of \( W^{I1} \), \( W^{I2} \), and \( W^{I3} \), respectively.

\[ \delta^0_{i} = (y_d,i(k) - y_i(k)) \] (46)
\[ \delta^b_j = \sum_{i=1}^{m}(\delta^0_{i} w^{I3}_{ij} f'(\cdot)) \] (47)
\[ \frac{\partial x_j(k)}{\partial w^{I1}_{jl}} = f'(\cdot)x_j(k - 1) + \frac{\partial x_j(k - 1)}{\partial w^{I1}_{jl}} \] (48)

**Weighted information criterion (WIC)**

The weighted information criterion (WIC) was initially proposed to find the best ANN model (Egrioglu et al. 2008). In this paper, WIC was applied to four real-time series datasets in order to measure the forecasting performance of the examined sub-models’ architectures and to decide the sub-models of the combined model. The number of input nodes depends on the number of sub-models of the combined model. The mean absolute percentage error (MAPE), root mean square error (RMSE), Akaike information criterion (AIC), Bayesian information criterion (BIC), and direction accuracy (DA) were chosen as the model selection criteria. The MAPE and RMSE were used to detect deviations between the actual values and the forecasting values. The AIC and BIC were used for penalizing large models. The DA measured the forecasting direction accuracy. Although the MAPE and RMSE can also be used individually to select a model, the models they select tend not to be sufficiently detailed. These criteria are calculated as follows:

\[ \text{MAPE} = \frac{1}{T} \sum_{i=1}^{T} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \] (49)
\[ \text{RMSE} = \sqrt{\frac{1}{T} \sum_{i=1}^{T} (y_i - \hat{y}_i)^2} \] (50)
\[ \text{AIC} = \log \left( \frac{\sum_{i=1}^{T}(y_i - \hat{y}_i)^2}{T} \right) + 2m \] (51)
\[ \text{BIC} = \log \left( \frac{\sum_{i=1}^{T}(y_i - \hat{y}_i)^2}{T} \right) + \frac{m \log(T)}{T} \] (52)
\[ \text{DA} = \frac{1}{T} \sum_{i=1}^{T} a_i, a_i = \begin{cases} 1 & \text{if } (y_{i+1} - \hat{y}_i)(\hat{y}_{i+1} - y_i) > 0 \\ 0 & \text{otherwise} \end{cases} \] (53)

where \( y \) is the actual value, \( \hat{y} \) is the forecasted value, \( T \) is the total number of data items, and \( m \) is the number of ANN weights.

In this paper, we used a special criterion called the modified direction accuracy (MDA), which was proposed according to the special direction accuracy criterion. The MDA criterion is calculated in the following way:

\[ A_j = \begin{cases} 1 & y_{j+1} - y_j \leq 0 \\ 0 & y_{j+1} - y_j > 0 \end{cases} \] (54)
\[ F_j = \begin{cases} 1 & \hat{y}_{j+1} - \hat{y}_j \leq 0 \\ 0 & \hat{y}_{j+1} - \hat{y}_j > 0 \end{cases} \] (55)
\[ D_j = (A_j - F_j)^2 \] (56)
\[ \text{MDA} = \frac{\sum_{j=1}^{T-1} D_j}{T - 1} \] (57)

The following describes the algorithm of model selection strategy based on WIC:

1. All of the structure of the sub-models, which consist of the combined model are determined. For instance, we have five input layer nodes, one output layer node, and 15
hidden layer nodes. Thus, the total number of possible structures is 18.

(2) The best weights of AIC, BIC, RMSE, MAPE, DA, and MDA are determined using the training data and calculated with the training data.

(3) The five criteria must be standardized for neural network structures.

(4) Then, the WIC is calculated as follows:
\[
WIC = 0.2 \times (\text{MAPE} + \text{RMSE}) + 0.1 \times (\text{AIC} + \text{BIC}) + 0.2 \times (\text{MDA} + (1 - \text{DA})).
\]

(5) We choose an architecture with the minimum WIC.

### The theory of the combined model

The combination forecasting theory indicates (Bates and Granger 2001) that if the M forecasting models can solve a certain forecasting problem, the weight coefficients should be appropriately selected, and then the results of the M forecasting methods are added to obtain a new model named the combined model. The results of the combined model are better than the M models. Assuming that the actual time series data is presented in the form of \( y_t \), the number of sample points is \( i \), \( y_i \) is the forecasting value obtained by the \( i \)th forecast model, the forecast error is \( e \), and the weight coefficient of the \( i \)th forecasting model is \( w \), then the general combined forecast model can be expressed as follows:

\[
y_t = \sum_{i=1}^{M} \omega_i (\hat{y}_i + e_i), \quad t = 1, 2, \ldots, L, \quad (58)
\]

\[
\hat{y}_t = \sum_{i=1}^{M} \hat{\omega}_i \hat{y}_i, \quad t = 1, 2, \ldots, L, \quad (59)
\]

where \( \hat{\omega}_i \) is the estimated value of \( \omega_i \) and represents the weight of each single model, \( \hat{y}_i \) is the forecasting value of the combined model. Determining the weight coefficient of each model is a key step in establishing a combined forecasting model. Then, by solving the optimization problem of the combination model, an optimal combination model can be obtained. Then this optimization problem can be expressed as:

\[
\text{Min} \sum_{i=1}^{L} |y_i - \hat{y}_i|, \quad (60)
\]

\[
\text{s.t.} \sum_{i=1}^{M} \omega_i = 1, 0 \leq \omega_i \leq 1, i = 1, 2, \ldots, M.
\]

When the predefined absolute error or the maximum number of iterations are reached, the optimization process is stopped.

### Multi-objective optimization theory

The optimization problems of the objective function with multiple measurement indexes in the domain of definition can be solved by multi-objective optimizations. The objective function checks and balances the shortcomings of the forecast model in many aspects by assigning weights to each measurement index, to improve the forecasting accuracy and stability. Generally speaking, multi-objective optimization problems (MOPs) can be divided into two categories: constrained problems and non-constrained problems. A constrained problem with \( j \) inequality and \( k \) equality constraints can be expressed as (Laumanns et al. 2006):

\[
\text{Minimize} \quad F(x) = (f_1(x), f_2(x), \ldots, f_M(x))^T
\]

\[
\text{s.t.} \quad g_j(x) \geq 0, j = 1, 2, \ldots, J, \quad x \in \Omega \quad (61)
\]

\[
h_k(x) = 0, k = 1, 2, \ldots, K,
\]

where \( M \) is the number of objectives, \( x = (x_1, x_2, \ldots, x_n)^T \) is the decision vector, and \( n \) is the number of decision variables. In (61), \( \Omega = \prod_{i=1}^{n} [x_{L}^i, x_{U}^i] \subseteq R^n \) is called the decision space, where \( x_{L}^i \) and \( x_{U}^i \) are the lower and upper limits of the decision variables, respectively.

When the inequality and equality constraints in (61) are omitted, an unconstrained multi-objective problem is obtained, which is expressed as follows

\[
\text{Minimize} \quad F(x) = (f_1(x), f_2(x), \ldots, f_M(x))^T
\]

\[
\text{s.t.} \quad x \in \Omega. \quad (62)
\]

And this method was widely used in many fields like carbon dioxide emissions forecast (Liu et al. 2022), electricity price forecast (Fu et al. 2020) and electricity demand forecast (Zhang et al. 2018).

### Acknowledgements

This work was supported by Western Project of the National Social Science Foundation of China (Grant No.18XTJ003).

### Funding

The work is funded by 'Western Project of the National Social Science Foundation of China (Grant No.18XTJ003).

### Data availability

Data not available due to legal and commercial restrictions.

### Declarations

#### Conflict of interest

The authors declare that they have no conflict of interest.

### References

Abdollahzade M, Miranian A, Hassani H, Iranmanesh H (2015) A new hybrid enhanced local linear neuro-fuzzy model based on the optimized singular spectrum analysis and its application for
Wind speed forecast based on combined theory, multi-objective optimisation, and sub-model selection

Egrioglu E, Aladag CH, Güney S (2008) A new model selection approach, part I: solving problems with box constraints. IEEE Trans Evol Comput 12(4):362–377

Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601

Contreras J, Espinola R, Nogales F, Conejo A (2003) ARIMA models applied to hourly wind speed and wind power. J Wind Eng Indus Aerodyn 49:319–328

Chang W-Y (2014) A literature review of wind forecasting methods. Power Energy Eng 2:161–168

Contreras J, Espinola R, Nogales F, Conejo A (2003) ARIMA models applied to predict next-day electricity prices. IEEE Trans Power Syst 18(3):1014–1020

Damousis IG, Alexiadis MC, Theocaris JB, Dokopoulos PS (2004) A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation. IEEE Trans Energy Convers 19(2):352–361

Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601

Diebold FX, Mariano R (1995) Comparing predictive accuracy. J Bus Econ Stat 20(1):134–144

Dorvolo AS, Jervase JA, Al-Lawati A (2002) Solar radiation estimation using artificial neural networks. Appl Energy 71(4):307–319

Diebold FX, Mariano R (1995) Comparing predictive accuracy. J Bus Econ Stat 20(1):134–144

Elman JL (1990) Finding structure in time. Cogn Sci 14:179–211

Enrique R, David B, Jorge-Juan B, Ana P (2019) Review of wind energy technology and associated market and economic conditions in Spain. Renew Sustain Energy Rev 101:415–427

Fried L, Qiao L, Sawyer S (2021) Global wind report, global wind energy council. https://gwecc.net/members-area-market-intelligence/gwc estratégias/

Fu T, Zhang S, Wang C (2020) Application and research for electricity price forecasting system based on multi-objective optimization and sub-models selection strategy. Soft Comput 24(20):15611–15630

Gers FA, Schmidhuber J (2000) Recurrent nets that time and count. Ieee-Ints-Enns Int Jt Conf Neural Netw 3:189–194

Global Wind Energy Council. Global wind statistics (2019), p. 2019

Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501

Inan G, Göktepe AB, Ramyar K et al (2007) Prediction of sulfate expansion of PC mortar using adaptive neuro-fuzzy methodology. Build Environ 42(3):1264–1269

Jang J-SR (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23(3):665–685

Laumanns M, Thiele L, Zitzler E (2006) An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. Eur J Oper Res 169(3):932–942

Lei M, Shiyan L, Chuanwen J, Hongling L, Yan Z (2009) A review on the forecasting of wind speed and generated power. Renew Sustain Energy Rev 13(4):915–920

Li G, Shi J (2010) On comparing three artificial neural networks for wind speed forecasting. Appl Energy 87(7):2313–2320

Liu M, Ling YY (2003) Using fuzzy neural network approach to estimate contractors’ markup. Build Environ 38(11):1303–1308

Liu Z, Jiang P, Wang J, Zhang L (2012) Ensemble system for short term carbon dioxide emissions forecasting based on multi-objective tangent search algorithm. J Environ Manag 302:113951

Meng K, Yang H, Dong ZY, Guo W, Wen F, Xu Z (2016) Flexible operational planning framework considering multiple wind energy forecasting service providers. IEEE Trans Sustain Energy 7(2):708–717

Neshat M, Adeli A, Sepidnam G (2012) Predication of concrete mix design using adaptive neural fuzzy inference systems and fuzzy inference systems. Int J Adv Manuf Technol 63(1–4):373–390

Niu T, Wang J, Zhang K, Du P (2018) Multi-step-ahead wind speed forecasting based on optimal feature selection and a modified bat algorithm with the cognition strategy. Renew Energy 118:213–229

Riahy G, Abedi M (2008) Short term wind speed forecasting for wind turbine applications using linear prediction method. Renew Energy 33:35–41

Schwenker F, Kestler HA, Palm G (2001) Three learning phases for radial-basis- function networks. Neural Netw 14(4–5):439–458

Sftos A (2000) A comparison of various forecasting techniques applied to mean hourly wind speed time series. Renew Energy 21(1):23–35

Shamshad A, Bawadi M, Hussain WW, Majid T, Sanusi S (2005) First and second order Markov chain models for synthetic generation of non-stationary wind speed series. Energy 30(5):693–708

Smith DA, Mehta KC (1993) Investigation of stationary and nonstationary wind data using classical Box-Jenkins models. J Wind Eng Indus Aerodyn 49:319–328

Somans SS, Zareipour H, Malik O et al (2010) A review of wind power and wind speed forecasting methods with different time horizons. In: North American Power Symposium (NAPS). IEEE, 2010. 1–8.

Specht DF (1991) A general regression neural network. IEEE Trans Neural Netw 2(6):568–576

Torres JL, Garca A, Blas MD, DeFrancisco A (2005) Forecast of hourly average wind speed with arma models in navarre (Spain). Sol Energy 79(1):65–77

Vapnik V (1997) The nature of statistic learning theory. Springer, Berlin

Wang X, Sideratos G, Hatzigiannou N et al (2004) Wind speed forecasting for power system operational planning. In: International conference on probabilistic models applied to power systems. IEEE, pp 470–474

Wang J, Zhang W, Wang J et al (2014) A novel hybrid approach for wind speed prediction. Inf Sci 273:304–318

Wang S, Zhang N, Wu L et al (2016) Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method. Renew Energy 94:629–636
Wang JZ, Yang WD, Du P, Niu T (2018) A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm. Energy Convers Manag 163:134–150

Wang C, Zhang S, Xiao L, Fu T (2021) Wind speed forecasting based on multi-objective grey wolf optimisation algorithm, weighted information criterion, and wind energy conversion system: a case study in Eastern China. Energy Convers Manage 243:114402

Wu J, Hsu C, Chen H (2009) An expert system of price forecasting for used cars using adaptive neuro-fuzzy inference. Expert Syst Appl 36(4):7809–7817

Xiao L, Wang J, Dong Y, Wu J (2015) Combined forecasting models for wind energy forecasting: a case study in China. Renew Sustain Energy Rev 44:271–288. https://doi.org/10.1016/j.rser.2014.12.012

Xiao L, Dong Y, Dong Y (2018) An improved combination approach based on Adaboost algorithm for wind speed time series forecasting. Energy Convers Manage 160:273–288

Yan J, Li F, Liu Y, Gu C (2017) Novel cost model for balancing wind power forecasting uncertainty. IEEE Trans Energy Convers 32(1):318–329

Yang Y, Chen Y, Wang Y, Li C, Li L (2016) Modelling a combined method based on ANFIS and neural network improved by DE algorithm: a case study for short-term electricity demand forecasting. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2016.07.053

Yu C, Li Y, Zhang M (2017) An improved wavelet transform using singular spectrum analysis for wind speed forecasting based on elman neural network. Energy Convers Manag 148:895–904

Zhang W, Qu Z, Zhang K, Mao W, Ma Y, Fan X (2017) A combined model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting. Energy Convers Manag 136:439–451

Zhang S, Wang J, Guo Z (2018) Research on combined model based on multi-objective optimization and application in time series forecast. Soft Comput. https://doi.org/10.1007/s00500-018-03690-w

Zhang S, Wang C, Liao P, Xiao L, Fu T (2022) Wind speed forecasting based on model selection, fuzzy cluster, and multi-objective algorithm and wind energy simulation by Betz’s theory. Expert Syst Appl 116509

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.