Received: 1 January 2024 | Revised: 5 January 2025 | Accepted: 11 January 2025

IET Renewable Power Generation

DOI: 10.1049/rpg2.70002

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Enhancing wind power forecasting and ramp detection using long short-term memory networks and the swinging door algorithm

Ravi Pandit[1] | **Shikun Mu**[2] | **Davide Astolfi**[3]

[1]School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, Cranfield, UK

[2]Department of Computer Science, University of Exeter, Exeter, UK

[3]Department of Information Engineering, Università degli Studi di Brescia, Brescia, Italy

Correspondence
Ravi Pandit, School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, Cranfield, UK.
Email: ravi.pandit@cranfield.ac.uk

Abstract

Accurate prediction of short-term wind power ramps is essential for effective smart grid management. This study introduces the swinging door algorithm for ramp detection, which outperforms traditional methods by precisely identifying ramp events. Additionally, a long short-term memory (LSTM) network is evaluated against established models such as support vector machines, artificial neural networks, convex multi-task feature learning, and random forest for wind power ramp forecasting. The LSTM model demonstrates superior performance, achieving the lowest weighted mean absolute percentage error of 8.36% and normalized root mean squared error of 0.60, alongside the highest *R*-squared (R^2) value of 0.73, indicating strong predictive accuracy and correlation with observed data. Furthermore, the combined swinging door algorithm-LSTM framework improved ramp event detection by 15% compared to traditional methods, showcasing its robustness in capturing both mild and extreme ramp events. This research underlines LSTM's effectiveness in wind power forecasting, marking a notable advancement in prediction methodologies. By illustrating the strengths of LSTM and swinging door algorithm, the study contributes to the refinement of prediction models for smart grid applications, highlighting their potential to transform wind power ramp prediction and detection.

1 | INTRODUCTION

Wind power has emerged as the fastest-growing renewable energy source globally, playing a crucial role in diversifying electricity mixes in various countries. For example, in 2021, Germany's wind power contribution reached 25.6% of its electricity mix [1], while India's installed capacity expanded to 39.2 GW [2]. China, a leader in renewable energy, added 22.4 GW in the same year, totalling 281.5 GW in installed wind power capacity [3]. However, the sector's rapid growth introduces challenges in maintaining reliable operations and effective real-time control of power systems. The integration of wind power introduces uncertainties, especially due to large-scale ramp events that cause significant variations in energy output. The ramp events in wind power refers to the rapid change in wind power output over a short period. This change is usually caused by sudden variations in wind speed or other meteorological conditions (e.g. cold air intrusion, thunderstorms). The phenomenon can manifest as a sharp increase in power output (positive ramp) or a sharp decrease (negative ramp). Wind power ramp events cause grid frequency and voltage fluctuations, increasing reserve capacity stress and system instability risks. Frequent power changes accelerate wear on equipment and storage systems, shortening their lifespans. These events also lead to economic losses, such as higher operational costs and reduced renewable energy utilization. Accurately predicting these events is crucial and involves various models such as long short-term memory networks (LSTM), support vector machines (SVM), artificial neural network (ANN), convex multi-task feature learning (CMTFL), and random forests. The effectiveness of ramp event prediction is further enhanced by integrating these forecasts with the swinging door algorithm (SDA).

This study contributes to wind power forecasting by systematically comparing multiple indices from prevalent prediction models to identify the most efficient approach. Our analysis indicates that the LSTM model excels in forecasting wind power, and its integration with the SDA algorithm improves ramp event detection accuracy. This combination enables more reliable wind power output predictions, facilitating better energy demand management.

The paper is structured as follows: Section 2 reviews existing research in wind power forecasting and ramp extraction. Section 3 outlines the data pre-processing steps. Section 4 introduces the swinging door algorithm and the binary ramp definition. Section 5 presents and compares various forecasting models including LSTM, SVM, random forest, ANN, and CMTFL. Section 6 displays the prediction results for each model, with a comprehensive comparison and data analysis in Section 7, culminating in the conclusions.

2 | RELATED LITERATURE REVIEWS

Recent years have seen a growing focus on predicting wind power ramp events, an area that has diversified into wind power prediction and ramp event detection [4]. Within wind power prediction, the traditional methodologies fall into three main categories: physical, statistical, and machine learning-based models [5]. Physical models, leveraging data from numerical weather prediction (NWP) systems, predict wind power using meteorological and environmental inputs but often lack local precision. Statistical models, such as ARIMA, Bayesian learning, and Gaussian processes [4], excel in short-term forecasting by mining data to establish predictive relationships. Machine learning techniques, on the other hand, are recognized for their proficiency in classification, feature extraction, and nonlinear regression tasks within wind energy forecasting [6].

This paper primarily focuses on an advanced application of long short-term memory (LSTM) networks, particularly a modified LSTM structure that has shown promise in various wind turbine applications [7, 8]. Our modified LSTM includes dual peepholes to improve the forgetting gate and activation function, addressing the limitations of conventional feature selection methods that often compromise computational efficiency. The paper also explores the application of Bi-LSTM networks for long-term prediction, considering their potential beyond short-term forecasting [10].

Regarding ramp event detection, the Binary Definition algorithm is commonly referenced in literature [11, 12]. A novel addition to this field is the swing gate algorithm introduced by He [13], alongside Fu et al.'s visualization-based model [14]. Our work differs in that it juxtaposes various machine learning methods for wind power prediction and ramp event identification. We have identified the LSTM model and the SDA as the most effective combination, integrating SDA's ramp event recognition with LSTM's wind power prediction. This integration enables the LSTM model to incorporate values from identified ramp events in its training process, enhancing both short-term and long-term prediction capabilities through cell augmentation's optimization model.

3 | DATA AND METHODOLOGIES

3.1 | Data description and pre-processing

The dataset used in this study is sourced from a wind turbine supervisory control and data acquisition (SCADA) system deployed in Turkey for power generation. The SCADA system records various parameters, including wind speed, wind direction, and power generation, at 10-min intervals. The dataset comprises five parameters: date, active power, wind speed (m/s), theoretical power curve (KWh), and wind direction (°). To focus on the key factors influencing power generation, namely wind speed, theoretical power curve, and wind direction, the data was selected based on correlation analysis.

During the pre-processing step, samples with missing or negative power values were filtered out. Additionally, data points where the maximum wind speed exceeded 20 m/s, representing the turbine's cut-off wind speed, were removed, as the turbine ceases operation beyond this threshold. The original dataset contained 50,530 rows, and after pre-processing, it retained 32,429 rows. The pre-processed dataset was then split, using 90% (29,186 data points) for model training and reserving the remaining 10% (3243 data points) for prediction evaluation. The data pre-processing primarily inspired by [13, 14] which includes steps such as removing samples with negative or missing power values, as well as data points exceeding the cut-off wind speed of 20 m/s, to ensure the quality and reliability of the dataset for model training and evaluation. This resulted significant dropped of data during the pre-processing steps. To investigate whether the general descriptions of the data remain consistent after cleaning, we conducted a thorough analysis of the dataset's distribution of different features before and after pre-processing. We examined key statistical measures, such as mean, standard deviation, and range, to assess the impact of data cleaning on the data distribution.

Our investigation reveals that the general descriptions of the data remained largely consistent after cleaning. The distribution of wind speed, theoretical power curve, and active power exhibited similar patterns before and after pre-processing. We observed minimal variations in statistical measures, indicating that the pre-processing steps did not significantly alter the underlying characteristics of the dataset.

In the context of our study, the relationship between key meteorological variables and wind power generation, referred to as 'active power', is quantitatively assessed using three Pearson correlation coefficients: R1, R2, and R3. These coefficients are pivotal in understanding the linear associations between these variables and Active Power.

- **Pearson correlation coefficient R1**: This coefficient evaluates the linear relationship between wind speed and Active Power. It is a measure of how changes in wind speed correspond to variations in power output. A higher R1 value indicates a stronger positive correlation, signifying that increases in wind speed are typically associated with increases in Active Power.
- **Pearson correlation coefficient R2**: R2 assesses the correlation between air temperature and active power. This metric is crucial as it provides insights into how variations in air temperature are linearly related to changes in wind power output. Temperature influences wind patterns, which in turn can affect the efficiency of wind power generation.

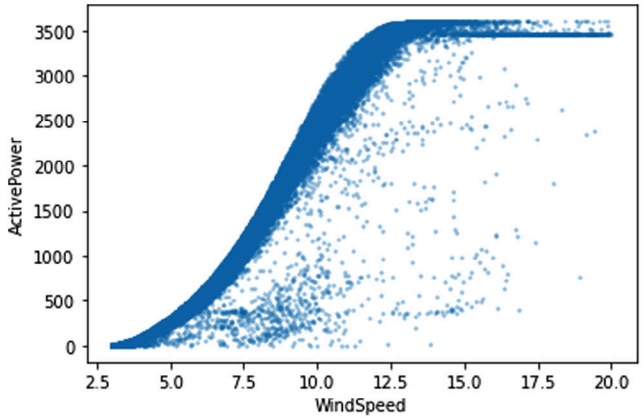

FIGURE 1 Wind turbine power curve correlation.

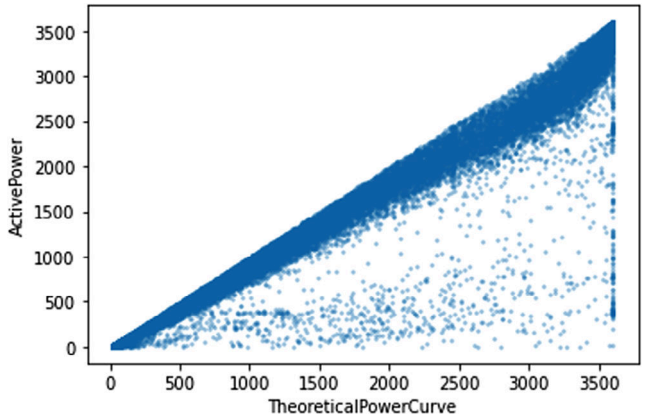

FIGURE 2 Theoretical power curve and active power.

- **Pearson correlation coefficient R3**: This coefficient is used to analyse the correlation between wind direction and active power. It highlights how different wind directions may impact the efficiency of wind turbines in generating power.

Additionally, the correlations between wind speed, wind direction, theoretical power curve, and active power were examined using the Pearson correlation coefficient. The obtained correlation values (R1 = 0.94, R2 = 0.97, and R3 = 0.02) were used to assess the relationship between these variables and active power. While wind speed and theoretical power curve exhibited strong correlations with active power, wind direction demonstrated a significantly smaller correlation, indicating its limited influence on active power. As a result, wind direction was deemed negligible during model training for wind power forecasting. The first two factors exhibit a strong relationship with active power, consistent with the observations in Figures 1 and 2. However, R3 is significantly small, suggesting that wind direction has limited influence on active power and can be neglected during model training. Hence, for wind power forecasting, we will employ active power, wind speed, and theoretical power curve at 10-min intervals as input variables to predict future wind power. Furthermore, we performed a comparative analysis of the correlations between wind speed, theoretical power curve, and wind direction with active power, both before and after data cleaning. The correlation coefficients (R1, R2, and R3) remained consistent, reaffirming the strength of the relationships between these variables and active power.

The correlations between wind speed, wind direction, theoretical power curve, and active power were assessed using the Pearson correlation coefficient. The results revealed strong correlations between wind speed and theoretical power curve with active power. However, wind direction exhibited a significantly smaller correlation, indicating its limited influence on active power. As a result, wind direction was deemed negligible during model training for wind power forecasting.

Additionally, the correlations between wind speed, wind direction, theoretical power curve, and active power were examined using the Pearson correlation coefficient. The obtained correlation values (R1 = 0.94, R2 = 0.97, and R3 = 0.02) were used to assess the relationship between these variables and active power. While wind speed and theoretical power curve exhibited strong correlations with active power, wind direction demonstrated a significantly smaller correlation, indicating its limited influence on active power. As a result, wind direction was deemed negligible during model training for wind power forecasting. The first two factors exhibit a strong relationship with active power, consistent with the observations in Figures 1 and 2. However, R3 is significantly small, suggesting that wind direction has limited influence on active power and can be neglected during model training. Hence, for wind power forecasting, we will employ active power, wind speed, and theoretical power curve at 10-min intervals as input variables to predict future wind power. Finally, the categorical feature "Date" was transformed using cyclical encoding. This method preserves the periodicity of the dates and enables the models to capture seasonal patterns effectively. The cyclical encoding allows the "Date" feature to be represented as sine and cosine transformations, ensuring compatibility with the machine learning models used in this study. This approach addresses the impact of the categorical feature on wind power forecasting and ramp event detection, enabling a comprehensive analysis of the dataset.

3.2 | Proposed methodologies

The methodology employed for wind ramp event prediction is presented in Figure 3. Initially, the acquired dataset undergoes a data cleaning process to eliminate anomalous data points, including negative power values and out-of-range values, as discussed in the previous section. Subsequently, 90% of the data (comprising 29,186 data points) is allocated for training purposes, while the remaining 10% serves as the testing dataset (containing 3243 data points). These datasets are utilized for constructing both the swinging door algorithm (SDA) and long short-term memory (LSTM) models.

Furthermore, a comparison is conducted between the SDA model and the conventional binary recognition method to demonstrate the effectiveness of the SDA approach. The training data is then employed to train the LSTM model for wind power prediction. Additionally, other models such as support vector machine (SVM), random forest, ANN, and CMTFL

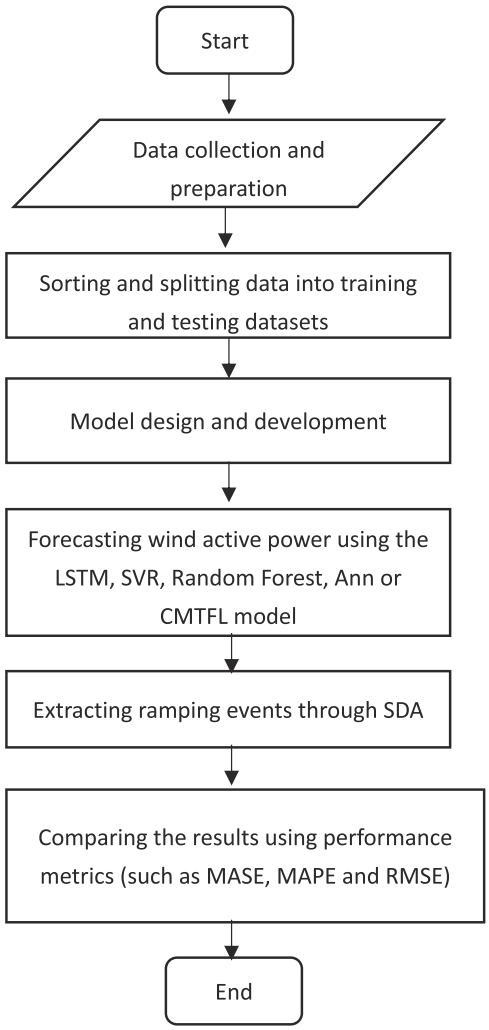

FIGURE 3 Flow chart of ramp event prediction model.

feature learning models are trained using the same data. Subsequently, the results of each model's wind power prediction, as well as the identification of ramp events, are compared by analysing the outcomes against the test data. Finally, the effectiveness of the models is evaluated based on performance indices, which will be discussed in a subsequent section.

4 | WIND RAMP EVENT DETECTION

To identify ramp events from the existing wind data set, we have compared several different ramp event detection methods and found the most suitable one. A brief description of the methods used in this paper follows.

4.1 | Swinging door algorithm (SDA)

In this work, we adopted an algorithm from the field of data compression, called the swinging door algorithm (SDA), that is used to identify ramp events from wind and solar energy. The SDA calculation and structure are simple, and only one parameter is required in its definition [15]. Indeed, the whole SDA compression and decompression process can be carried out in real-time. The only adjustable parameter is ε, the width of a "door" in the algorithm, which directly allows the (threshold) sensitivity to noise and insignificant fluctuations to be specified. The SDA enhances ramp detection by dynamically adjusting to data variations, offering a more robust approach compared to traditional methods such as binary definition and fixed threshold-based techniques. Unlike static methods that rely on predetermined thresholds and struggle to differentiate between significant events and transient noise, SDA utilizes a dynamic "swinging door" mechanism. This mechanism defines a flexible range around each data point, allowing the algorithm to filter out minor fluctuations while identifying meaningful ramp events when the data crosses outside the range. The use of a single adjustable parameter, ε (epsilon), enables the algorithm to adapt its sensitivity to noise, thereby minimizing false positives and negatives. In Figure 4, the swing door algorithm is briefly illustrated. (1) starting from an initial point on the y-axis (the black point at $t = 0, y = 5$) and defining a threshold gate of width ε on the y-axis; (2) obtaining a new point A and drawing auxiliary lines from both sides of the "hinge of the gate" to point A; (3) selecting points in the same way until a lower threshold point D is obtained. The upper door line and the lower door line are now parallel. Furthermore, the new iteration will still start from point D [15, 16].

Given the wind power interval (i, j) for all discrete time points and the objective function J of the dynamic programming model, the wind power ramp event (WPRE) can be detected by maximising the objective function:

$$J(i,j) = \max_{i<k\le j} \left[S(i,k) + J(k,j)\right], i < j \tag{1}$$

Where $J(i,j)$ is the maximum score in the interval (i,j), which can be calculated as the maximum over $(i–j)$ subproblems. The term $S(i,k)$ is a positive score value corresponding to the interval (i,k), which conforms to a super-additivity property:

$$S(i,j) > S(i,k) + S(k+1,j) \quad \forall i < k < j \tag{2}$$

$$S(i,j) = (j-i)^2 * R(i,j) \tag{3}$$

Where $R(i,j)$ defines a ramp in the interval (i,j).

It is worth to note that the objective variable in our study is the prediction of wind power ramp events, defined as significant changes in wind power output within a specified time frame. To achieve this, we utilized a range of explanatory variables that included wind speed, wind direction, air density, and temperature.

4.2 | Ramp event binary definition

4.2.1 | Greaves

According to [17], a ramp is a change in power output that exceeds the minimum size, $S_{\min}$, within a duration less than or

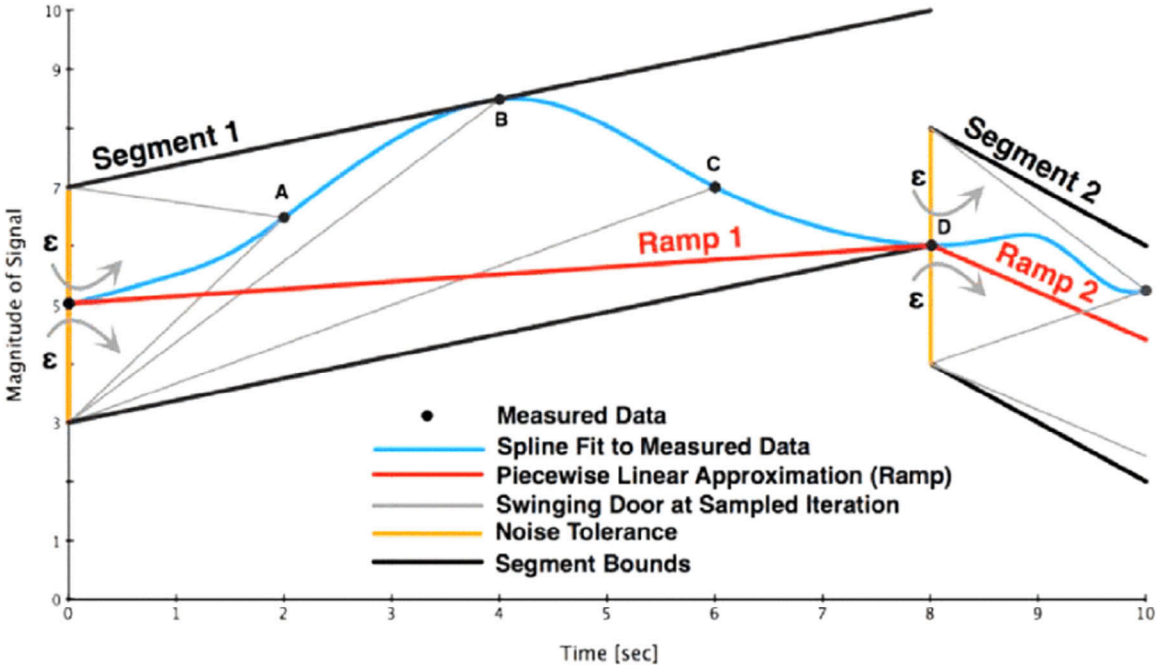

FIGURE 4 Swinging door algorithm performance [16].

equal to the maximum duration, $d_{\max}$. The ramp event is taken within 4 h or less to analyse the ramp prediction accuracy, and the power change must be 50% or more of the capacity [17].

4.2.2 | Bossavy

To solve the difficulty of normalizing ramp events due to parameter variations, Bossavy proposes a definition. Let $(p_t)_t$ be the wind time series, $(p_t^f)_t$ be the relevant filtered signal:

$$p_t^f = \text{mean}\left\{p_{t+h} - p_{t+h-n_{\text{am}}}; h = 1, \ldots, n_{\text{am}}\right\} \quad (4)$$

where n_{am} stands for the number of averaged differences of measures. The filtered signal $(p_t^f)_t$ measures the variations of the initial power signal $(p_t)_t$.

In Bossavy's work [18], the time t for which the filtered signal p_t^f has a maximum value is associated with a ramp event. The number of averaged measures n_{am} is the width of the filter f_{nam} and hence can be understood as a smoothing parameter. Small values of n_{am} will make the filtered signal $(p_t^f)_t$ more sensitive to short-period variations of the power output $(p_t)_t$. Bossavy chooses to use $\Gamma = 25\%$ of the nominal power of the wind farm and a value of $n_{\text{am}} = 5$ h [18].

4.2.3 | Bradford

Bradford defines a ramp event as a change in wind power output of greater than or equal to 20% capacity in magnitude over 1 h. In this article, Bradford sets a power percentage capacity (PPC), defined as:

$$\text{PPC} = \frac{\text{Turbine power output}}{\text{Maximum turbine power output possible}} \times 100 \quad (5)$$

Calculate each forecast hour. PPC differences more significant than or equal to 20% in magnitude for several consecutive hours are reported as ramp events. In Bradford's study, the time frame for ramp events was 2 h. If the PPC changes by more than 20% within this time frame, it is counted as a ramp event [19].

4.2.4 | Cutler

Cutler proposes a two-stage ramp detection method [20].

- Stage 1: Find all instances where the hourly averaged wind power changes by more than 75% of rated power within 3 h. Two individual hourly ramps occurring within 6 h are considered the same event.
- Stage 2: Removing the periods for where ramp events were found in stage 1; find all instances where the 10 min averaged power changes by more than 65% of rated power in 1 h. Two 10 min ramps are considered individual events if they occur more than 1 h apart. These are later combined if they occur roughly on the same day and categorised as a variable period.

4.3 | Comparison of ramp event detection methods

In this section, we compare various ramp event detection methods and evaluate their performance on selected data sets after

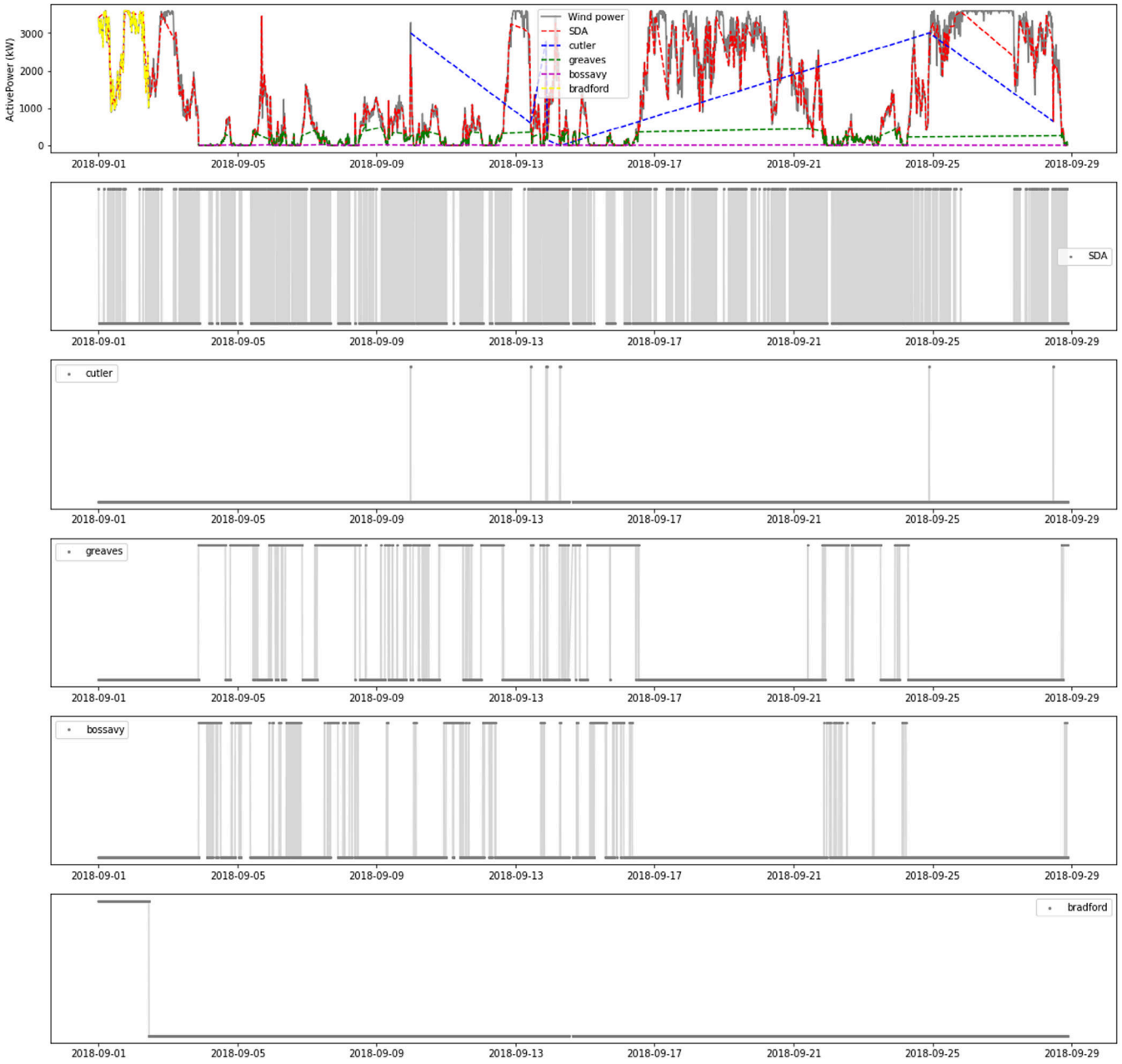

FIGURE 5 SDA and binary definition ramp event extraction.

data processing. The objective is to identify the most suitable method for processing the complete data set. Figure 5 illustrates the results of different methods in identifying ramp events for the month of September 2018 where, the grey line graph represents the active power at each time point in the original data, while lines in other colours indicate ramp events detected by different methods. In the five grey line graphs below, these ramp events and non-ramp events are separated by assigning them to different time points, with each grey line in the chart representing a single ramp event. This approach allows the line graph to intuitively display the number and time points of ramp events identified by different methods. Among these methods, the swinging door algorithm (SDA) is highlighted for its ability to identify the highest number of ramp events. In contrast, the Cutler and Bradford methods detect a significantly smaller number of ramp events. The figure, along with Table 1, offers a more detailed comparison, showing that while the Bradford method recognizes some ramp events, these often correspond to single events of extended duration. On the other hand, methods like Greaves' and Bossavy's detect multiple ramps but are not as extensive in their detection capabilities as the SDA.

To intuitively observe the ramp detection performance of different methods, we divided the September data into weekly

TABLE 1 SDA and binary definition ramp event comparison.

	Ramp events count	Mean duration (min)	Mean power change (KW)	Mean change percentage (%)
1st week				
SDA	10	58.9	1108.341	55.904
2nd week				
SDA	103	65	534	236.016
Cutler	1	190	194.184	93.5
Bossavy	36	26.8	267.043	85.8
Bradford	1	710	173.179	10.735
Greaves	32	92	310.142	25.5
3rd week				
SDA	111	56	502.496	302.7
Cutler	11	345	1498	594.6
Bossavy	25	177.6	304.98	132.5
Bradford	54	10	122.46	63
Greaves	36	156	74.539	84.19
4th week				
SDA	72	133.5	982.306	148.6
Cutler	6	211.7	1954.93	111.67
Bossavy	11	204.5	451.85	94.33
Bradford	1	540	141.87	4.9
Greaves	14	60.7	34.64	97.6
5th week				
SDA	28	250.4	1055.54	331.2
Cutler	77	89.3	957.87	70.1
Bossavy	8	865	937.56	1410.9
Bradford	1	260	55.138	59.05
Greaves	5	40	3.202	495.18

intervals. Figure 6 presents the number of ramp events identified by the swinging door algorithm (SDA) across different weeks of September 2018. The data for September is divided into weekly intervals to observe the ramp detection performance more intuitively. Notably, the SDA detects some ramps in the first week, a period during which other methods failed to identify any ramp events. The figure also shows that the Bradford method tends to overfit in the second week, a trend that continues in subsequent weeks. The Cutler method, while performing well in ramp recognition for the three weeks following the first, shows fluctuating effectiveness. Similar variability in performance is observed with Bossavy's and Greaves' methods. Despite some promising results from the binary definition methods, the SDA demonstrates more stable and reasonable results overall. It's important to note that September starts midweek, so the first and fifth weeks are not complete, while weeks 2 to 4 are full weeks. Based on the performance evaluation with this sample data set, the swing door algorithm exhibits the most effective recognition performance. Therefore, it is selected to identify ramp events in the complete data set.

5 | FORECASTING WIND ACTIVE POWER MODELS

5.1 | Long short-term memory (LSTM) theoretical descriptions

The LSTM is an improved RNN model. Compared with the traditional RNN model, LSTM adds a memory unit to evaluate whether the information is valuable, which solves the problem of gradient disappearance and gradient explosion in the training process of long sequences. This work uses LSTM under the Tensorflow framework to predict ramp events in the wind power system. The basic unit structure of the LSTM is shown in Figure 7, including three control gates: input, forget and output gate. In addition, the unique memory unit of LSTM can retain important ramp features, and the forgetting gate can discard unimportant features to strengthen the network's ability to learn ramp features [21]. The following formula gives the cell state at time step t:

$$C_t = f_t \odot C_{t-1} + i_t \odot g_t \tag{6}$$

Where $\odot$ denotes the Hadamard product (element-wise multiplication of vectors) and C_{t-1} is the previous cell state value [21]. The hidden state at time step t is given by:

$$h_t = O_t \odot \sigma_c (C_t) \tag{7}$$

The activation function formula of each gate is:

$$\begin{cases} i_t = \sigma\left(W_{xiyt} + W_{hih_{t-1}} + W_{cic_{t-1}} + b_i\right) \\ f_t = \sigma\left(W_{xfyt} + W_{hih_{t-1}} + W_{cic_{t-1}} + b_f\right) \\ o_t = \sigma\left(W_{xoyt} + W_{hoh_{t-1}} + W_{coc_{t-1}} + b_o\right) \\ c_t = f_{tc_{t-1}} + i_t tanh\left(W_{xcyt} + W_{hch_{t-1}} + b_c\right) \\ h_t = \sigma_t\ tanh\,(c_t) \end{cases}$$

where σ is the sigmoid activation function or the tanh activation function; yt is the input vector at time t; W_{xi}, W_{hi}, W_{ci}, W_{xf}, W_{xo}, W_{ho}, W_{co}, W_{xc}, and W_{hc} are all weight parameter matrices; b_i, b_f, b_o, and b_c are all bias vectors; c_t is the instant state vector of the state unit; h_t is the output of the state unit; it is the output of the input gate; f_t is the output of the forget gate; and o_t is the output of the output gate [22].

In LSTM wind power forecasting model, several key hyperparameters were carefully selected to optimize performance. The activation functions used within the LSTM layers were typically 'tanh' for recurrent activation and 'sigmoid' for gate activations. The optimizer, a critical component affecting the model's convergence during training, was chosen from options like Adam or RMSprop, known for their suitability in recurrent neural networks. The learning rate, a crucial aspect in the training process, was set to determine the step size at each iteration while minimizing the loss function. For loss functions, we relied on mean squared error (MSE) or mean absolute error (MAE), suitable for regression tasks. The batch size, defining the number of training samples per iteration, was selected to balance training

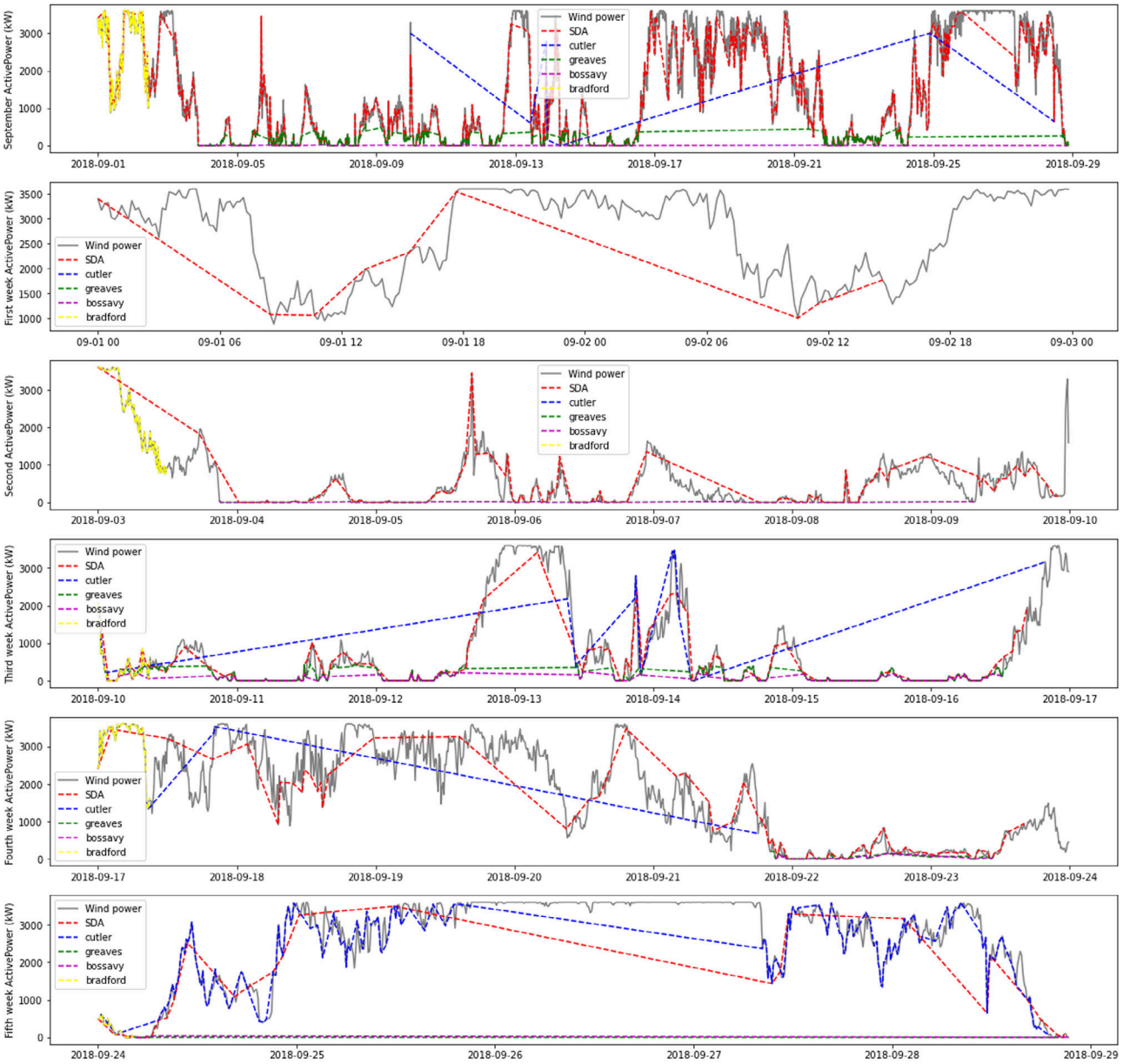

FIGURE 6 Ramp event corresponds to instant power generation (September starts midweek. So the first week and the fifth week are not complete weeks. Weeks 2 to 4 are full weeks).

speed and stability. The number of epochs was set to determine how many times the learning algorithm would process the entire training dataset. To identify the optimal hyperparameters, we utilized techniques like grid search or random search, systematically exploring various parameter combinations and evaluating their performance to select the most effective configuration based on validation dataset results.

5.2 | Support vector machine theoretical descriptions

SVM is a set of interrelated supervised learning methods for pattern recognition, regression, classification, estimation, and operator inversion [23]. The prediction task completed by SVR (support vector regression) on the input test case x is represented by the following equations:

$$\hat{y} = \sum_{i=1}^{n} \alpha_i k_{\text{rbf}}(x_i, x_*) + b \tag{8}$$

Where α_i is the weight, most time, it will be zero. N is the size of training data. $k_{\text{rbf}}(x_i, x_*)$ is the support-vector. In our article, we will use the RBF kernel.

From the above equation, the radial basis function (RBF) kernel is defined as:

$$k_{\text{rbf}}(x_p, x_q) = \exp\left[\frac{-(x_p - x_q)^2}{2\sigma^2}\right] \tag{9}$$

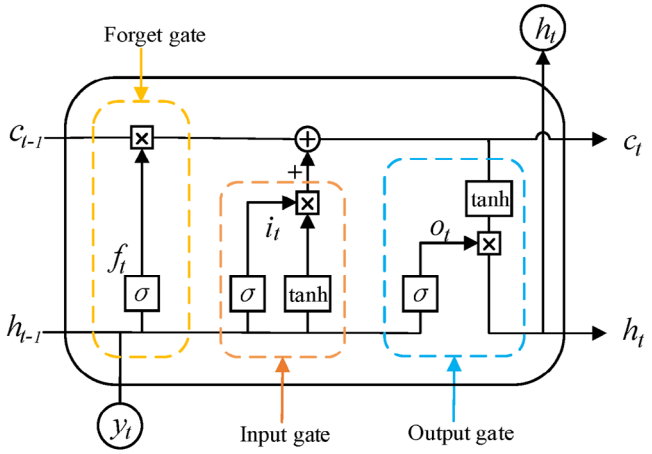

FIGURE 7 Simple illustration of LSTM model [22].

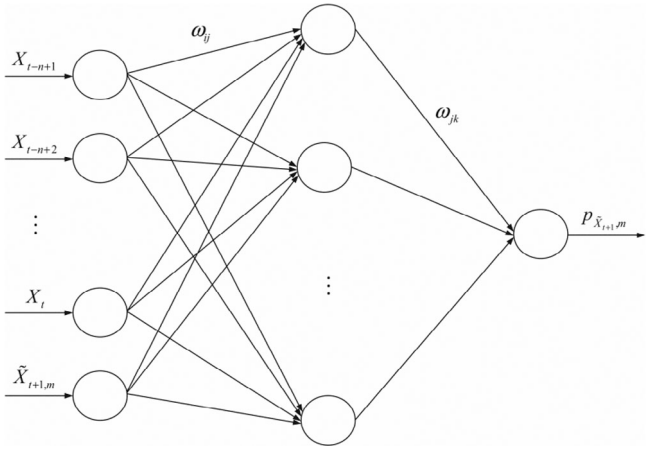

FIGURE 8 ANN model simple theoretical explanation diagram.

Using the grid search technique, we used this kernel to train the proposed SVM model and carried out hyperparameter optimisation.

5.3 | Random forest theoretical descriptions

Random forest is an ensemble technique developed by Breiman to build a set of trees based on a given training input [24]. Mathematically, the regression task is,

$$\hat{A} = \frac{1}{k}\sum_{i=1}^{k}\hat{r}(X, V_i) \tag{10}$$

Where $\hat{r}(X, V)$ is the representative tree at the end of the training process, X is the sum of a set of input feature vectors and V is the collective set of input–output pairs $V_i = (X_1, \ddot{y}_1), (X_2, \ddot{y}_2), \ldots (X_n, \ddot{y}_n)$. The predicted output is averaged over k in the decision tree. Another advantage of this method is that it is not sensitive to noise from uncorrelated trees through differential input sampling. Common problems of overfitting are avoided by adjusting hyperparameters, such as the maximum number of features and tree depth of the random forest model [24].

5.4 | ANN theoretical descriptions

The NN probability generation model is shown in Figure 8 [25] and applied as follows. Assuming that the current time is t, the input variable X contains $(n+1)$ inputs, where n inputs include current wind power (at time t) and historical wind power data (at time $t-1$, $t-2$,…, $t-n+1$). The output variable O at the next moment contains the probability $p_{\tilde{x}_{t+1,m}}$ of the mth possible wind value.

The probability generation model has $(n+1)$ input layer node, 1 is the output layer node, and l is the hidden layer node. There are a total of $[(n+1)l+l\times1]$ weight parameters and $(l+1)$ threshold parameters. The calculation process of the model is as follows. The network input and output $(X|O) = X_{t-n+1}, X_{t-n+2}, \ldots, X_{t-1}, X_t, \tilde{X}_{t+1,m}|_{p_{\tilde{x}_{t+1,m}}}$ needs to be determined. Hidden layer outputs H are calculated based on the input variables X, connection weight parameters ω_{ij} and threshold parameters α as follows:

$$h_j = f\left(\sum_{i=1}^{n+1}\omega_{ij}x_j - \alpha_j\right), j = 1, 2, \ldots, l \tag{11}$$

Where l is the hidden layer nodes; f is the hidden layer excitation function.

There are many types of excitation functions, and this paper uses the function as:

$$f(x) = \frac{1}{1+e^{-x}} \tag{12}$$

According to the hidden layer outputs H, connection weights ω_{jk} and threshold parameter β, the probability $p_{\tilde{X}t+1,m}$ of the mth possible wind power value at the next moment is calculated as the neural network output:

$$p_{\tilde{x}_{t+1,m}} = \sum_{j=1}^{l} h_j\omega_{ij} - \beta, k = 1 \tag{13}$$

$$p_{\tilde{x}_{t+1,m}} = \left\{p_{\tilde{x}_{t+1,1}}, p_{\tilde{x}_{t+1,2}}, \ldots, p_{\tilde{x}_{t+1,m}}, \ldots, p_{\tilde{x}_{t+1,M}}\right\} \tag{14}$$

$$\sum_{m=1}^{M} p_{\tilde{x}_{t+1,m}} = 1 \tag{15}$$

Where $p_{\tilde{x}_{t+1,m}}$ is the totality of all the probability values, and M is the total number of possible wind power values at the next moment [25].

5.5 | Convex multi-task feature learning (CMTFL)

Inspired by Argyriou's et al. work [26], the author proposes a method based on a non-convex regularizer to learn sparse

representation shared from multiple tasks. The algorithm has a simple explanation: it alternately performs supervised and unsupervised steps. In the previous step, it learns task-specific functions. The last step learns general cross-task sparse representations of these functions.

6 | RESULTS AND DISCUSSIONS

This section presents the results of applying the wind power forecasting models described earlier. The SDA was selected for ramp event identification, and its results were used as the criterion for selecting the wind power forecasting method. We compared the LSTM model with other methods and demonstrated its effectiveness and adaptability based on the identification results of SDA.

Figure 9 illustrates the wind speed data and wind power forecast obtained using the LSTM method. The first row shows the wind speed data for December 2018, while the second row displays the ramps detected using the SDA method. Identified ramp events are indicated with a value of 1, representing a ramp event, while 0 denotes a standard power generation time point. The LSTM model predicts these ramps, as shown in the third row. It is evident that the wind power forecast closely matches and fits the source data. Additionally, there is a significant overlap between the ramp events identified by the source data using the swing door algorithm and the ramp events identified through the LSTM model.

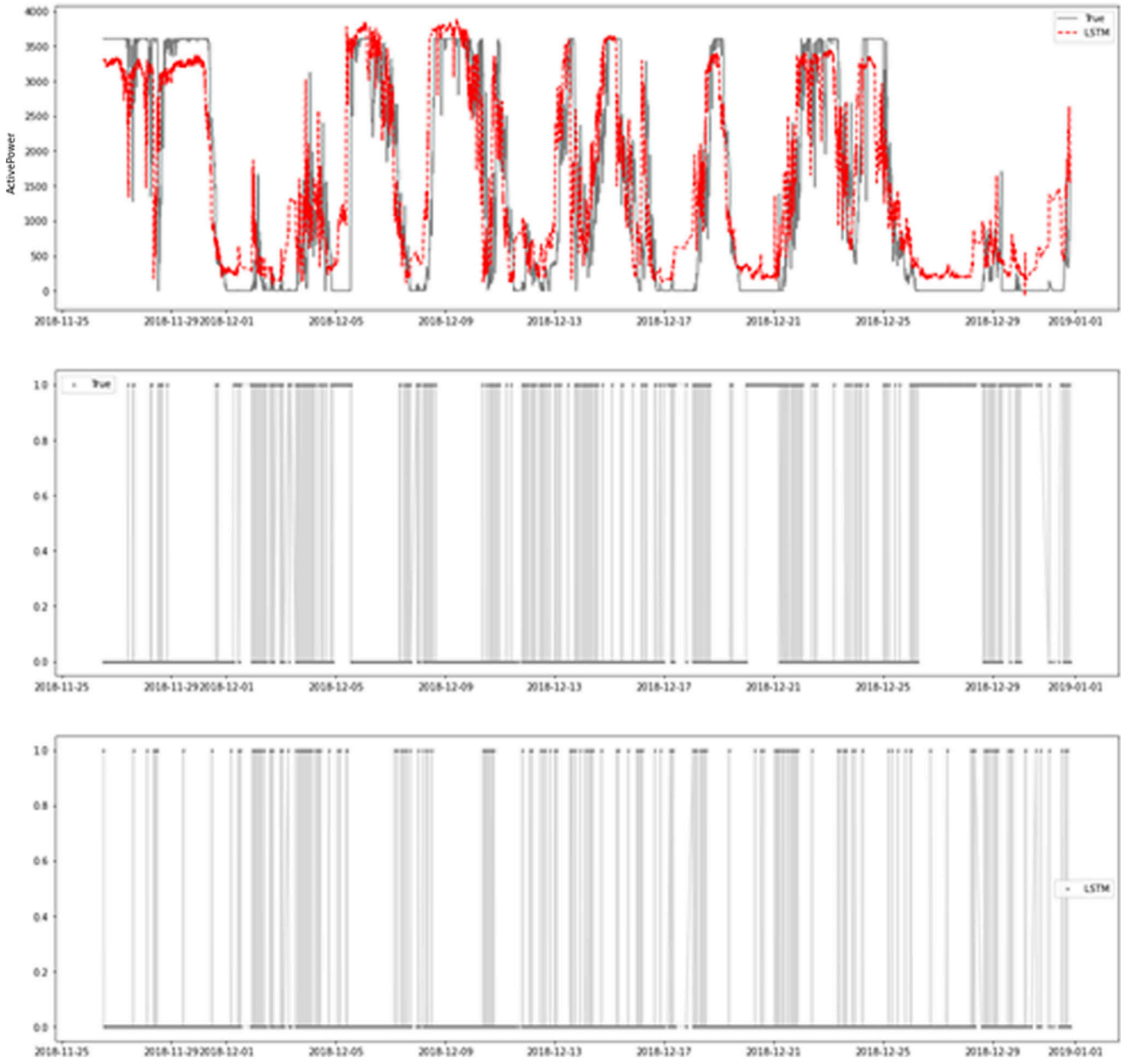

FIGURE 9 Forecast wind power and ramp events through the LSTM model.

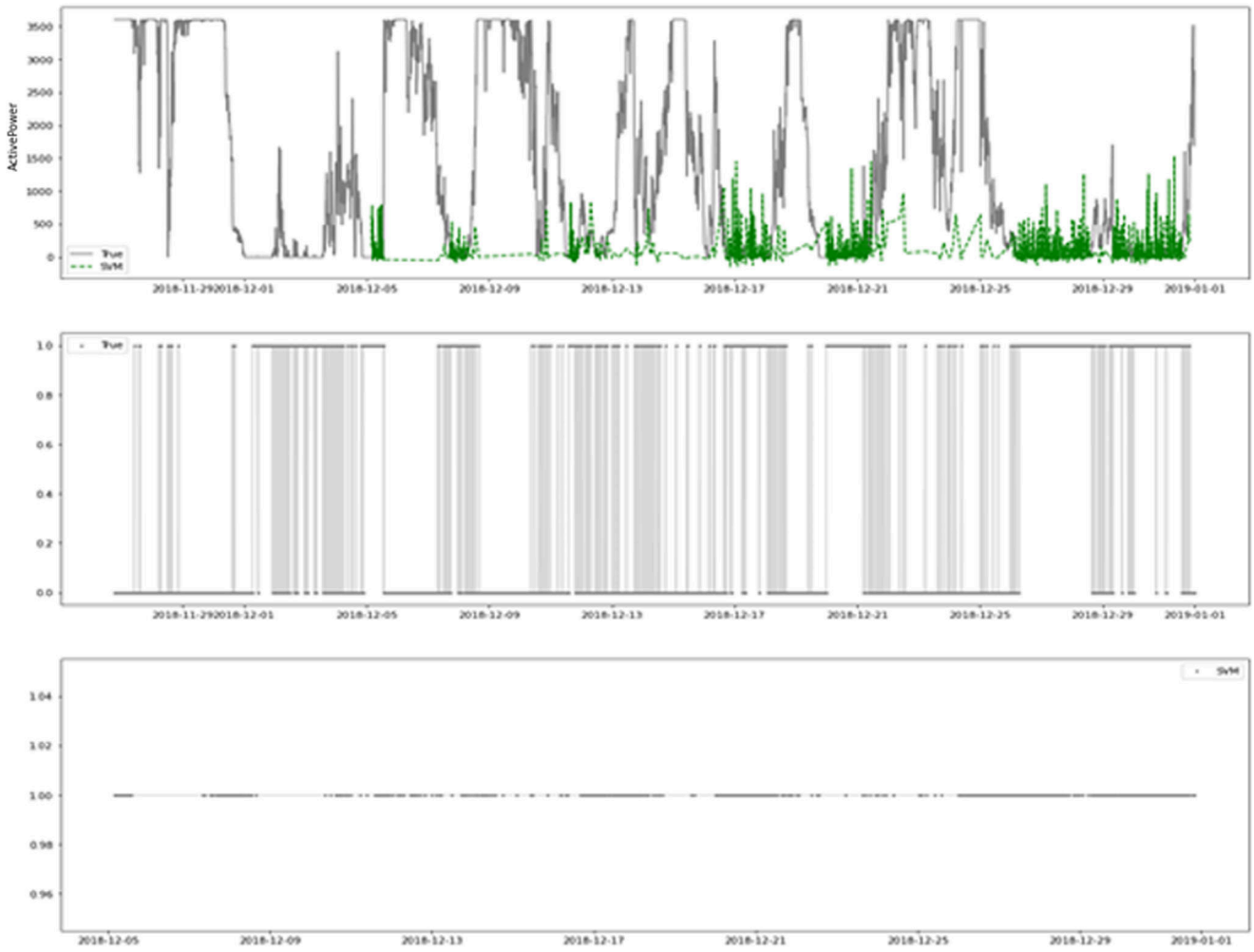

FIGURE 10 SVR model forecast wind power generation and ramp events.

Figure 10 presents the results obtained with the SVR model using the RBF kernel. The wind power prediction results of the SVR model mainly concentrate at lower levels, resulting in limited practical forecasting of wind power trends. While the model recognizes most ramp events, there is clear over-recognition apparent in the image display.

The wind power forecast simulated by the random forest (RF) model and the corresponding ramp events are shown in Figure 11. The random forest model successfully predicts ramp events, especially in the second half of the data set where numerous ramp events are identified. However, the overlap between the wind power forecast (WPF) and the wind power of the test data is insufficient. In several peak areas, the wind power simulated by Random Forest is lower than expected.

Figure 12 displays the power generation forecast and ramp event recognition obtained through the ANN model. The power generation forecast of the ANN model significantly deviates from the standard values, and the predicted values are relatively concentrated. The accuracy of ramp event recognition improves in the last 10 days, but the performance during the rest of the time is inconsistent with the original data.

We also achieved good ramp event prediction results using CMTFL, as shown in Figure 13. However, the fit between real-time power and the test data does not seem reasonable, with a delay observed between WPF and the actual wind power of the test group. The appropriateness of the model will be further discussed in a later section.

It is worth to note that, from Figures 9–13, grey line in the first figure represents the active power in source data, while the red line represents the simulated data generated by the random forest model. The second figure shows the ramp events in the source data, and the third figure depicts the ramp events identified in the simulated data using the model.

In the dataset we used, there were a total of 3219 instances of significant active power fluctuations. The LSTM model identified 1248 ramp events, the random forest model identified 359, the SVM model identified 478, the ANN model identified 653, and the CMTFL model identified 662. Figure 14 compares the ramp events identified by the original data with those identified by the five methods. The positions of the differently coloured dots in the image intuitively demonstrate the alignment between the time points of ramp events identified by different models

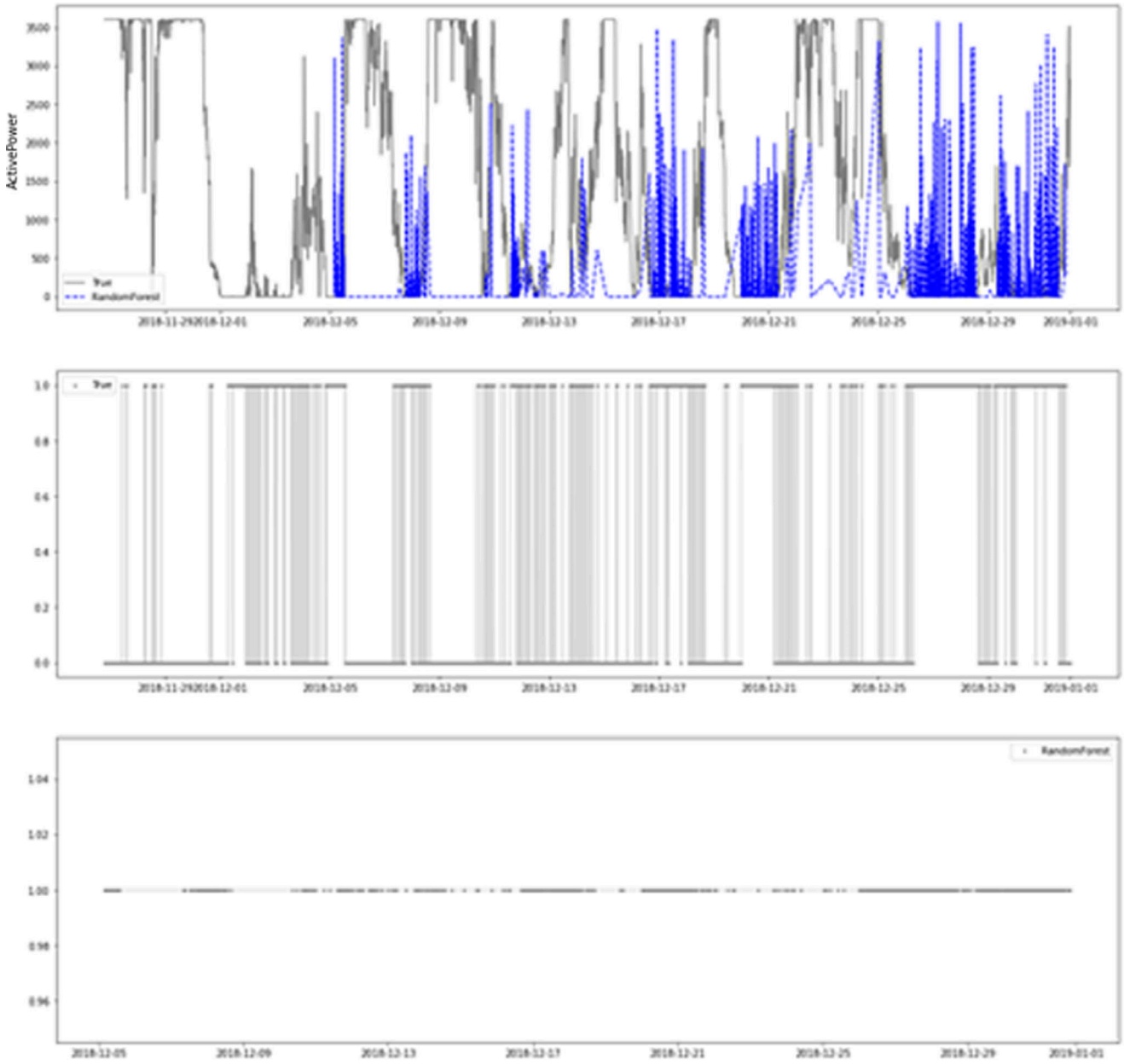

FIGURE 11 Random forest for wind power and ramp events detections.

and the fluctuation points in the source data. The grey dots at the top of the figure represent ramp events in the source data, while the coloured dots represent ramp events identified by different methods. The LSTM model demonstrates the best fit with the actual data in recognizing ramp events. This is evident from the alignment between the ramp events identified by LSTM and the original data. Although CMTFL includes most of the ramp events from the original data, it still falls short in accuracy as it recognizes ramp events in areas where the actual data does not indicate ramps. While the other three methods perform well in the second half of the recognition area, their accuracy, even in the last few days, is surpassed by LSTM. Overall, after careful consideration, we conclude that the LSTM model's prediction results are superior to those of other models.

7 | QUANTITATIVE COMPARISON BETWEEN PREDICTION METHODS

Although we can intuitively judge the strength and weakness of the model from the sample distribution, we still need to rely on data expression to ensure that our conclusions are correct. Some performance error metrics have been used to evaluate each method more accurately. Mean absolute error (MAE), mean absolute percent error (MAPE), weighted mean absolute percentage error (WMAPE), root mean square error (RMSE), normalized root mean squared error (NRMSE), and R-squared, are used to evaluate WPF performance. MAE and RMSE are used to evaluate the prediction ability and accuracy of the proposed prediction methods. In contrast, MAPE and R^2

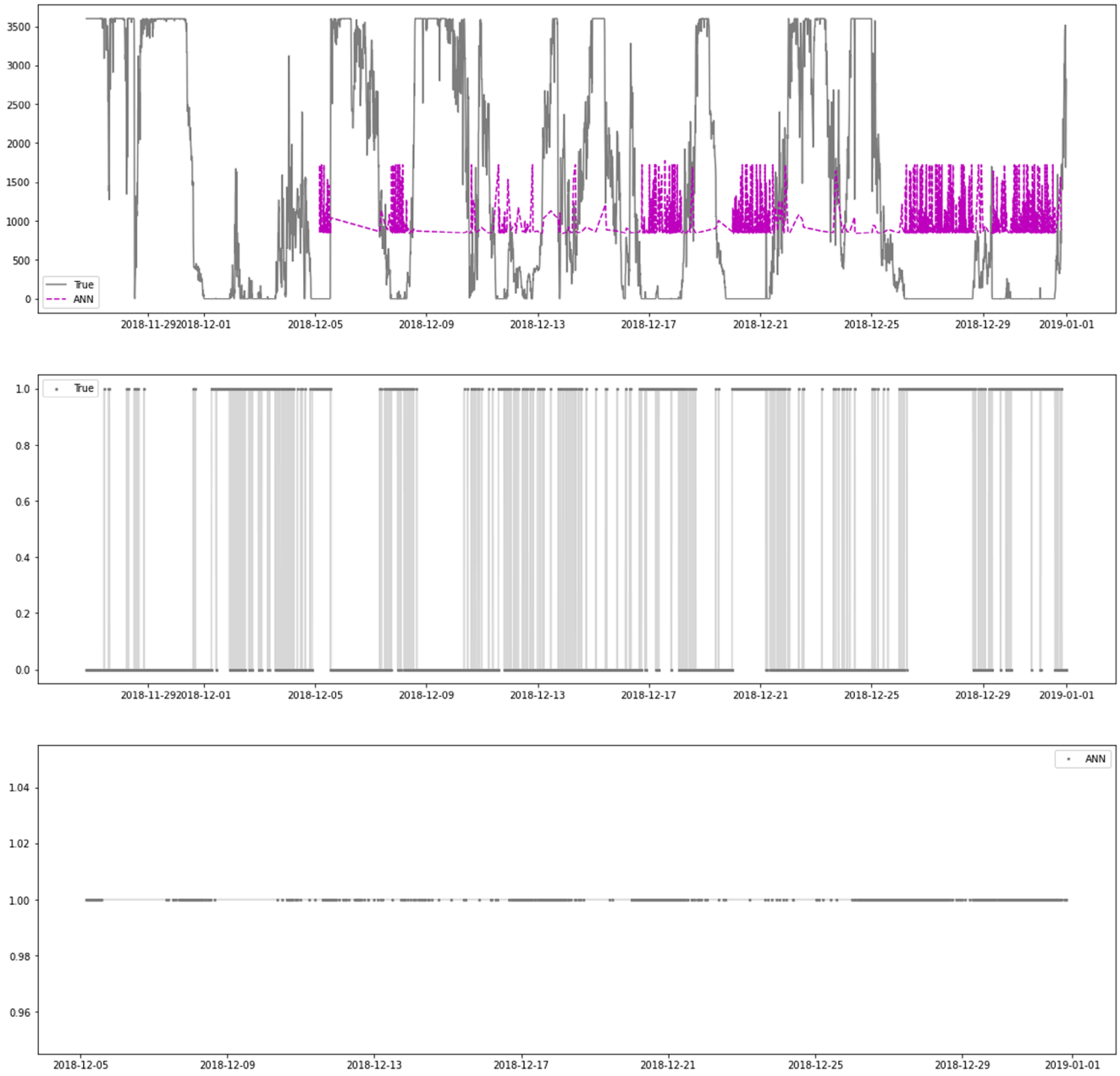

FIGURE 12 Forecast wind power and ramp events through ANN model.

assess the relative deviation between the point prediction results and the actual wind power output. WMAPE and NRMSE will demonstrate the adequate performance of these metrics. The mathematical definitions for these performance metrics are as follows [27, 28]:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|P_i^{\mathrm{F}} - P_i^{\mathrm{R}}\right| \tag{16}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|P_i^{\mathrm{F}} - P_i^{\mathrm{R}}\right|}{P_i^{R}} \times 100\% \tag{17}$$

$$\mathrm{WMAPE} = \frac{\sum_{t=1}^{n}|A_t - F_t|}{\sum_{t=1}^{n}|A_t|} \tag{18}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(P_i^{\mathrm{F}} - P_i^{\mathrm{R}}\right)^2} \tag{19}$$

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{\bar{y}} \tag{20}$$

$$R^2 = 1 - \frac{SS_{\mathrm{res}}}{SS_{\mathrm{tot}}} \tag{21}$$

The calculated values of these performance metrics are tabulated in Table 2 for WPF. Among the three indicators of MAE, RMSE and NRMSE, the LSTM model performed the best and was significantly better than other models. However, in MAPE, the LSTM model is not the best. The MAPE values of the SVM

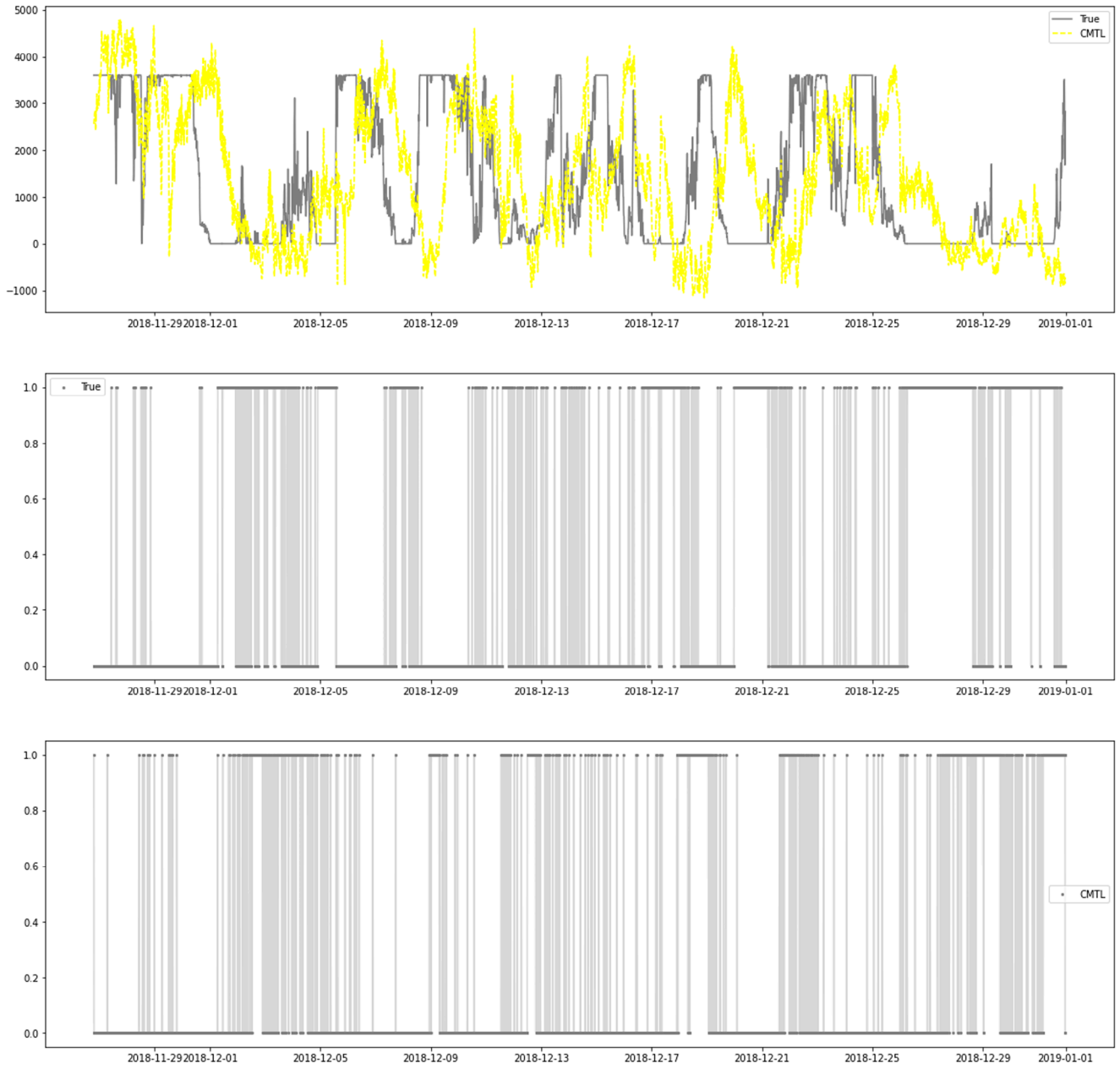

FIGURE 13 CMTFL model forecast wind power generation and ramp events.

TABLE 2 Comparative WPF errors.

Index	LSTM	SVM	Random forest	ANN	CMTFL
MAE	782.0	1104.7	1346.3	1279.27	1038.9
RMSE	1041.2	1340.8	1668.8	1517.2	1383.1
nRMSE	0.6034	0.8455	1.0523	0.9567	0.8721
MAPE	8.3659	7.3226	8.6445	11.8388	8.2341
WMAPE	0.4532	2259.10	2753.12	2616.05	1543.32
R^2	0.7261	0.7209	0.0188	0.3624	0.3041

model and the CMTFL model are both smaller than those of the LSTM model. However, there are many extreme values in the wind slope time. The numerical gap is not large, so we do not consider that this data reflects that the LSTM model has a weaker performance than SVM and CMTFL.

The data of WMAPE and R^2 can further reflect the superiority of the LSTM model. Looking at WMAPE, we can see that the metrics of the LSTM model are significantly better than the other models. It shows that the bridging effect of the LSTM model is the best. We can also see in Figure 15 that the predicted value of the LSTM model is in good agreement with the actual observed value of our experimental data. A higher R^2 value indicates that the variance of the model is similar to that of the true value. It shows how well the LSTM model fits the observations and how well it fits. A high R^2 indicates a strong correlation between observed and expected values.

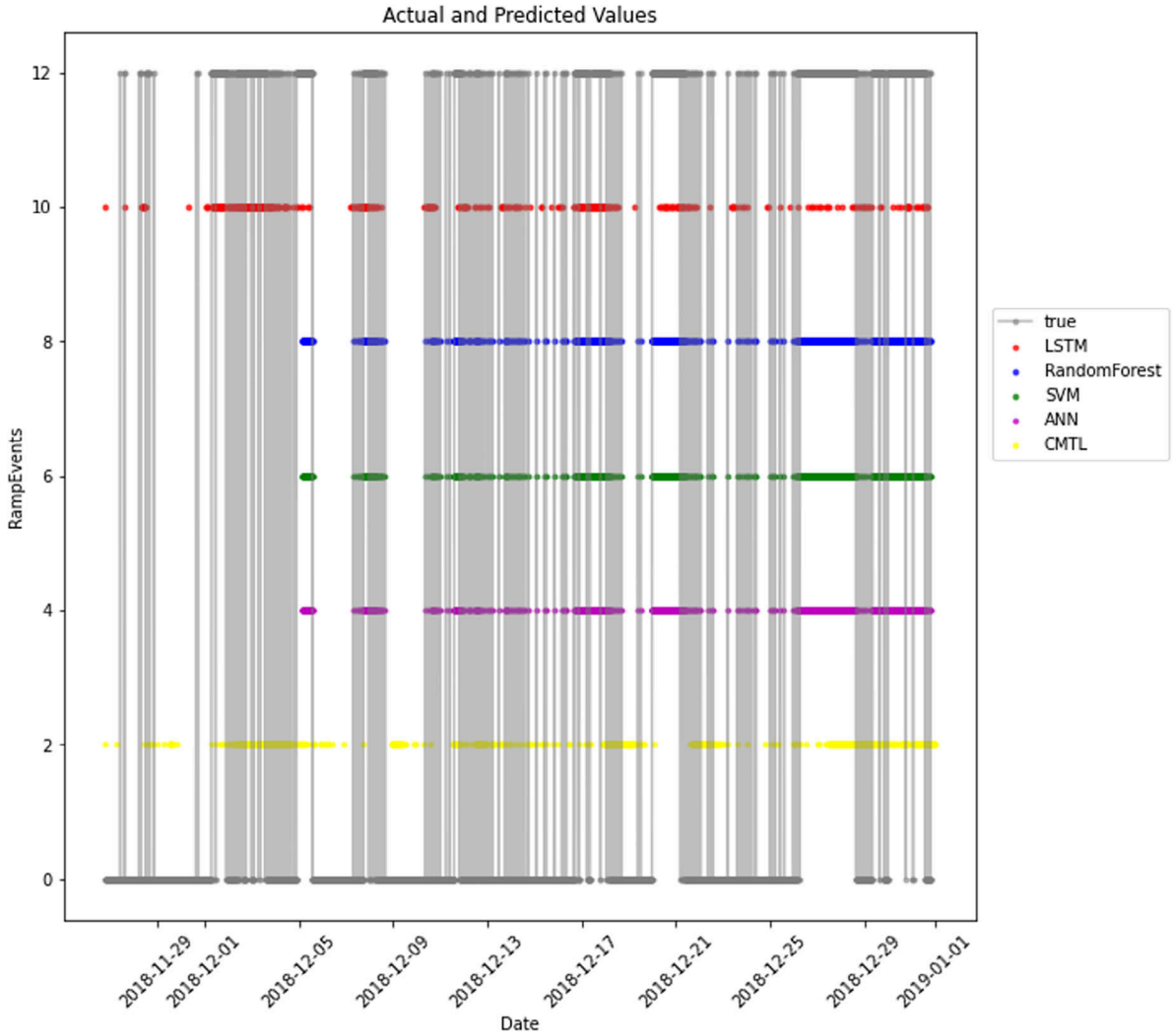

FIGURE 14 Extraction summary of ramp events of forecast results.

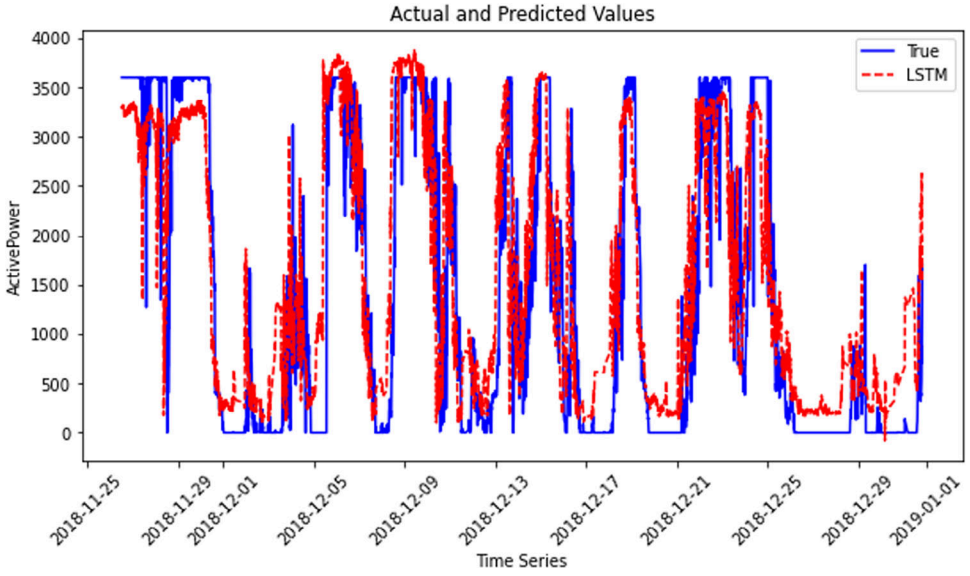

FIGURE 15 Comparison of wind power prediction value between test data and LSTM model.

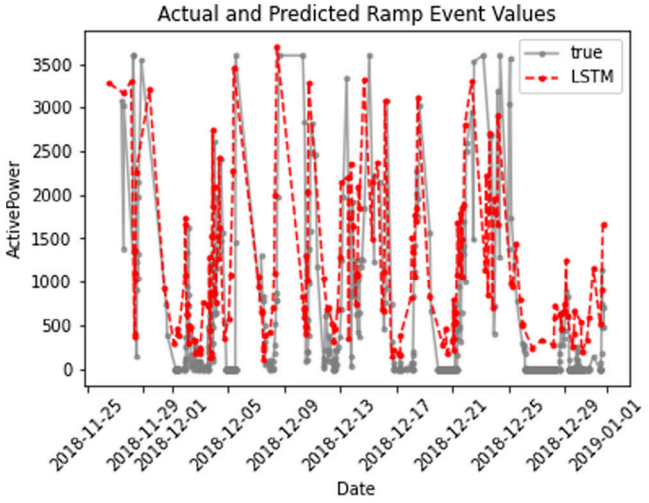

FIGURE 16 Comparison of wind power and ramp events between the test data and LSTM model.

After determining the reliability of the predicted data of the LSTM model, we can identify the ramp events by SDA. As shown in Figure 16, we can observe that the identification of ramp events forecasting fits well with the ramp events from the source data. Therefore, we can consider the LSTM model the most suitable prediction model. And by combining the LSTM model and SDA, we can achieve the purpose of predicting wind power ramping events.

It is worth to note that, although the SDA is effective in filtering out minor fluctuations and accurately identifying ramp events, it has certain limitations when applied to wind power ramp detection. One limitation is its sensitivity to the choice of the threshold parameter ε (epsilon), which directly affects the algorithm's performance. An improperly chosen ε value can result in either over-smoothing or over-segmentation, leading to missed ramp events or increased false positives. Additionally, SDA may struggle to detect ramps in highly volatile conditions where rapid changes in wind speed and power occur within short time intervals, as it may consider these changes as normal fluctuations. To address these limitations, this study employed a systematic approach to optimize the ε parameter using a grid search method based on minimizing error metrics such as MAE and RMSE.

8 | CONCLUSIONS AND FUTURE WORK

This study conducts an in-depth evaluation of various wind power prediction models and ramp event identification algorithms, ultimately identifying the SDA and LSTM models as the optimal combination. The SDA, in particular, shows remarkable effectiveness in ramp event classification, outperforming traditional binary methods. It utilizes generator power data to precisely identify ramp events, enhancing the accuracy of wind power forecasts.

For wind power prediction, we employed machine learning models coupled with time series analysis, using the SDA for extracting ramp events. The LSTM model stands out, demonstrating high accuracy in simulating ramp events within test data, as evidenced by low MAE values. Its robustness is particularly notable in scenarios where other models, like the SVM and Random Forest, falter, displaying significant discrepancies in WMAPE statistics. These models also struggle to comprehensively identify ramp events, unlike the LSTM, which accurately traces the general trend but still shows a measurable gap that suggests room for further refinement. A notable limitation of our study is the lack of differentiation between up-ramp and down-ramp events, a distinction crucial for detailed analysis. Future research should focus on identifying and integrating additional influential parameters to refine ramp event characterization.

Proposed model comparative performance with LSTM and other data-driven models (e.g. GRU, CNN) kept for future work. Nevertheless, proposed research findings highlight the necessity for ongoing optimization and the exploration of new parameters to increase the accuracy and depth of ramp event analysis in wind power forecasting.

AUTHOR CONTRIBUTIONS

Ravi Pandit: Writing—original draft; writing—review and editing; Formal analysis; investigation; project administration; resources; supervision. **Shikun Mu**: Conceptualization; data curation; formal analysis; investigation; methodology; software; validation; visualization. **Davide Astolfi**: Investigation; supervision; writing—review and editing.

ACKNOWLEDGEMENTS

There is no funding associated with this research. The views expressed are those of the authors alone.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Data available on request from the authors.

ORCID

Ravi Pandit https://orcid.org/0000-0001-6850-7922
Davide Astolfi https://orcid.org/0000-0002-8409-0298

REFERENCES

1. Renewables – global energy review 2021 – analysis. https://www.iea.org/reports/global-energy-review-2021/renewables (2021). Accessed 1 Jul 2023
2. The critical link to India's clean energy transition. https://gwec.net/wp-content/uploads/2021/06/GWEC-MEC-India-2021-10-June.pdf (2021). Accessed 1 Jul 2023
3. China: installed wind power capacity 2021. https://www.statista.com/statistics/950342/china-accumulated-installed-wind-power-capacity/ (2021). Accessed 1 Jul 2023
4. Lyners, D., Vermeulen, H., Groch, M.: Wind power ramp event detection using a multi-parameter segmentation algorithm. Energy Rep. 7, 5536–5548 (2021)
5. Hu, J., Zhang, L., Tang, J., Liu, Z.: A novel transformer ordinal regression network with label diversity for wind power ramp events forecasting. Energy 280, 128075 (2023)

6. Mishra, S., Bordin, C., Taharaguchi, K., Purkayastha, A.: Predictive analytics beyond time series: predicting series of events extracted from time series data. Wind Energy 25(9), 1596–1609 (2022)
7. Pandit, R.K., Kolios, A., Infield, D.: Data-driven weather forecasting models performance comparison for improving offshore wind turbine availability and maintenance. IET Renewable Power Gener. 14, 2386–2394 (2020)
8. Pandit, R., Astolfi, D., Tang, A.M., Infield, D.: Sequential data-driven long-term weather forecasting models' performance comparison for improving offshore operation and maintenance operations. Energies 15, 7233 (2022)
9. Du, M.: Improving LSTM neural networks for better short-term wind power predictions. In: 2019 IEEE 2nd International Conference on Renewable Energy and Power Engineering (REPE), pp. 105–109. IEEE, Piscataway, NJ (2019)
10. Moharm, K., Eltahan, M., Elsaadany, E.: Wind speed forecast using LSTM and Bi-LSTM algorithms over Gabal El-Zayt wind farm. In: 2020 International Conference on Smart Grids and Energy Systems (SGES), pp. 922–927. IEEE, Piscataway, NJ (2020)
11. Ouyang, T., Zha, X., Qin, L., Xiong, Y., Huang, H.: Wind power ramps prediction method based on amendment of similar events. Proc. CSEE 37(2), 572–580 (2017)
12. Gallego, C., Cuerva, Á., Costa, A.: Detecting and characterising ramp events in wind power time series. J. Phys. Conf. Ser. 555, 012040 (2014)
13. He, Y., Zhu, C., An, X.: A trend-based method for the prediction of offshore wind power ramp. Renewable Energy 209, 248–261 (2023)
14. Fu, J., Ni, Y., Ma, Y., Zhao, J., Yang, Q., Xu, S., Zhang, X., Liu, Y.: A visualization-based ramp event detection model for wind power generation. Energies 16, 1166 (2023)
15. Cui, Y., He, Y., et al.: Algorithm for identifying wind power ramp events via novel improved dynamic swinging door. Renewable Energy 171, 542–556 (2021)
16. Florita, A., Hodge, B.-M., Orwig, K.: Identifying wind and solar ramping events. In: 2013 IEEE Green Technologies Conference (GreenTech), pp. 147–152. IEEE, Piscataway, NJ (2013)
17. Greaves, B., Collins, J., Parkes, J., et al.: Temporal forecast uncertainty for ramp events. Wind Eng. 33(4), 309–319 (2009)
18. Bossavy, A., Girard, R., Kariniotakis, G.: Forecasting uncertainty related to ramps of wind power production. In: European Wind Energy Conference and Exhibition 2010, EWEC 2010, pp. 1099–1107. Curran Associates, Red Hook, NY (2010)
19. Bradford, K.T., Carpenter, R.L., Shaw, B.: Forecasting southern plains wind ramp events using the WRF model at 3-km. In: AMS Student Conference, pp. 1–10. American Meteorological Society, Boston, MA (2010)
20. Cutler, N., Kay, M., Jacka, K., Nielsen, T.S.: Detecting, categorising and forecasting large ramps in wind farm power output using meteorological observations and WPPT. Wind Energy 10(5), 453–470 (2007)
21. Obiora, C.N., Ali, A., Hasan, A.N.: Forecasting hourly solar irradiance using long short-term memory (LSTM) network. In: 2020 11th International Renewable Energy Congress (IREC), pp. 1–6. IEEE, Piscataway, NJ (2020)
22. Han, L., Qiao, Y., Li, M., Shi, L.: Wind power ramp event forecasting based on feature extraction and deep learning. Energies 13(23), 6449 (2020)
23. Pandit, R., Infield, D.: Comparative analysis of binning and support vector regression for wind turbine rotor speed based power curve use in condition monitoring. In: 2018 53rd International Universities Power Engineering Conference (UPEC), pp. 1–6. IEEE, Piscataway, NJ (2018)
24. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Routledge, London (2017)
25. Cui, M., Ke, D., Gan, D., Sun, Y.: Statistical scenarios forecasting method for wind power ramp events using modified neural networks. J. Mod. Power Syst. Clean Energy 3(3), 371–380 (2015)
26. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Mach. Learn. 73(3), 243–272 (2008)
27. Astolfi, D., Pandit, R., Gao, L., Hong, J.: Individuation of wind turbine systematic yaw error through SCADA data. Energies 15, 8165 (2022)
28. Astolfi, D., Pandit, R., Lombardi, A., Terzi, L.: Diagnosis of wind turbine systematic yaw error through nacelle anemometer measurement analysis. Sustainable Energy Grids Networks 34, 101071 (2023)

How to cite this article: Pandit, R., Mu, S., Astolfi, D.: Enhancing wind power forecasting and ramp detection using long short-term memory networks and the swinging door algorithm. IET Renew. Power Gener. 19, e70002 (2025). https://doi.org/10.1049/rpg2.70002

APPENDIX A: HYPERPARAMETERS OF LSTM AND SDA ALGORITHMS

This appendix provides detailed information on the hyperparameters used in the long short-term memory (LSTM) network and the swinging door algorithm (SDA) for our study on wind power forecasting and ramp detection.

A.1 | Long short-term memory (LSTM) network hyperparameters

The LSTM model was configured with the following hyperparameters:

1. Number of LSTM layers: 3
   - This includes one input layer, one hidden layer, and one output layer.
2. Number of units per Layer:
   - Input layer: 50 units
   - Hidden layer: 100 units
   - Output layer: 1 unit (for regression output)
3. Activation functions:
   - Recurrent activation: hyperbolic tangent (tanh)
   - Gate activations: sigmoid function
4. Optimizer: Adam
   - Learning rate: 0.001
5. Loss function: mean squared error (MSE)
6. Batch size: 32
7. Number of epochs: 100

A.2 | Swinging door algorithm (SDA) configuration

The SDA was configured with the following parameters:

1. Compression ratio: 0.1
   - This parameter determines the degree of data compression in the SDA.
2. Tolerance angle: 5 degrees
   - The tolerance angle sets the threshold for data deviation before a new "door" is created.
3. Minimum and maximum Duration Threshold:
   - Minimum duration: 10 min
   - Maximum duration: 60 min
   - These thresholds define the minimum and maximum time windows for detecting ramp events.

A.3 | Reproducibility note

The above hyperparameters were selected based on preliminary experiments and literature review. Researchers aiming to replicate or extend this study are advised to consider tuning these hyperparameters based on their specific datasets and computational resources.