Renewable Energy 230 (2024) 120830

Contents lists available at ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

Accelerating optimal scheduling prediction in power system: A multi-faceted GAN-assisted prediction framework

Ali Peivand, Ehsan Azad Farsani *, Hamid Reza Abdolmohammadi

Electrical and Computer Engineering Group, Golpayegan College of Engineering, Isfahan University of Technology, Golpayegan, 87717-67498, Iran

ARTICLE INFO

Keywords:
Optimal scheduling prediction
Generative adversarial network
Two-stage optimization
Wind power prediction
Optimal dispatched wind power

ABSTRACT

This study introduces a comprehensive framework aimed at enhancing power system optimality through a two-stage optimization process and the development of a deep-based model for optimal scheduling prediction (OSP). Initially, a Bidirectional Long Short-term Memory (Bi-LSTM) architecture is employed to accurately forecast wind power in the first stage. Subsequently, a convolutional Generative Adversarial Network (GAN) model utilizes these predicted wind power values to generate synthetic scenarios. These scenarios, based on the preceding 10 days' wind power predictions, serve as inputs for the subsequent power system optimization stage. To streamline computational efficiency, the power system optimization is conducted via a two-stage model. The outputs from this process, alongside other pertinent parameters, are utilized to train the proposed deep-based OSP model. The efficacy of the proposed model in rapidly and reliably predicting optimal scheduling is evaluated using the 118-bus power system. Results indicate that the innovative approach demonstrates exceptional speed and precision in determining optimal scheduling for the power system. Specifically, the proposed OSP model accurately forecasts optimal dispatch for ten days ahead in a mere 0.38 s, with an error rate below 0.001. Furthermore, the model exhibits a 92 % correlation in predicting optimal dispatched wind power. Sensitivity analysis highlights that optimizing the arrangement of the proposed deep-based model using an automatic hyperparameter optimization software framework (OPTUNA) can significantly enhance performance accuracy, potentially by up to 24 %.

Nomenclature

Acronyms:	
ANN	Artificial Neural Network
OPTUNA	An Automatic Hyperparameter Optimization Software Framework
Bi-LSTM	Bidirectional Long Short-term Memory
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DT	Decision Tree
GRU	Gated Recurrent Unit
KNN	K-Nearest Neighbor
LSTM	Long Short-term Memory
ML	Machine Learning
OSP	Optimal Scheduling Prediction
PSO	Particle Swarm Optimization
PyPSA	Python for Power System Analysis
RFR	Random Forest Regression
RNN	Recurrent Neural Network
UC	Unit Commitment
Indices:	

(*continued on next column*)

(*continued*)

t	Index of time	l	Index of layers
d	Index of generators	j	Index of dataset
γ	Index of loads		
MUT^d, MDT^d	Minimum uptime and downtime for dth generator, hour	M^d, CP^d	Marginal and capital costs for dth generator, $
SUC^d, SDC^d	Start-up and shutdown costs of dth generator, $	λ_{ls}	Load-shedding penalty coefficient, $/MWh
Parameters:			
Decision variables:			
$P_G^{d,t}$, $P_G^{d,nom}$	Dispatched and nominal power of dth generator at the time t, MW	RU^d, RD^d	Ramp-up and ramp-down limits of dth generator, MW
$P_G^{d,max}$, $P_G^{d,min}$	Maximum and minimum power limits of dth generator, MW	$C^{d,t}$	Commitment status of dth generator at the time t
$S_{startup}^{d,t}$, $S_{shutdown}^{d,t}$	Start-up and shutdown costs of dth generator at the time t, $	$P_{LS}^{\gamma,t}$	Load shedding power related to the γ^{th} load at the time t, MW

* Corresponding author.
E-mail address: e.azad@iut.ac.ir (E. Azad Farsani).

https://doi.org/10.1016/j.renene.2024.120830
Received 21 January 2024; Received in revised form 8 May 2024; Accepted 16 June 2024
Available online 19 June 2024

1. Introduction

The ever-growing presence of sustainable energy sources, such as wind power, poses some risks to the power system. For instance, China's total wind power capacity surged to 280 GW by the end of 2020 [1]. Also, based on the Danish government's goal, wind power will be slated to account for an impressive 50 % of the country's electricity consumption by 2025 [2]. This extensive capacity mandates a precise planning framework. In this situation, one of the common phenomena is wind curtailment, which leads to the waste of large amounts of clean power. Curtailment should be considered to guarantee the stable maintenance of the power system regarding the uncertain behavior of wind power. However, it tends to spill a significant amount of energy which can be diminished substantially provided that an effective framework is taken into account. Therefore, in the case of implementation of a proper plan, wind curtailment can be decreased crucially [3,4]. In this way, accurate prediction of wind power can be extremely beneficial as it provides the network operator with an optimal and accessible amount of wind power. Hence, ensuring meticulous prediction of wind power is essential from a power system perspective, and it is equally crucial to implement fast and reliable optimal scheduling. These can lead to wind curtailment reduction as well as economic dispatching.

To implement optimal scheduling for a power system with highly penetrated wind power, it is necessary to consider the uncertain behavior of wind power. One effective way to deal with the stochastic behavior of wind power is a meticulous prediction. Firstly, Machine Learning (ML) methods develop rough estimations of time-series variables with undeniable errors that are unable to capture all hidden relationships. Recent significant enhancements in Artificial Neural Networks (ANNs) have resulted in the emergence of some effaceable memory-aid methods which are mostly suitable to time-series variables, in this case wind power. By increasing the complexity and the number of hidden layers of the shallow MLs, a more efficient learning approach has emerged, namely Deep Neural Networks (DNNs) which are able to learn more complicated hidden patterns in observations. Various methods have been proposed in different references for wind forecasting modeling, while recurrent-based approaches such as Recurrent Neural Networks (RNNs), have better results due to their ability to remember previous records. However, RNNs can struggle with long-term and highly fluctuated data, and they may be prone to vanishing gradient issues in certain cases [5]. To surmount this issue, the Gated Recurrent Unit (GRU) structure has been proposed to mitigate the gradient-related issues. In the GRU structure, the model has been learning to filter out irrelevant information [6]. Meanwhile, reference [7] embellished the GRU structure with an attention mechanism for fitness promotion. A more qualified layer, the Long Short-term Memory (LSTM), is introduced by Ref. [8] as a substitute for the GRU layer which has more gates and parameters. Also, its intricate architecture ends up capturing more long-time dependencies in sequential data. Similar to the GRU structure, an attention layer can significantly augment the time-series prediction models. For instance, as proposed in Ref. [9], a two-stage attention--based LSTM model can reduce the error values by approximately 10 % compared to using LSTM alone. Also, the modified version of LSTM, named Bidirectional LSTM (Bi-LSTM) [10], used two LSTM layers in forward and backward directions to capture dependencies in the past and future. Consequently, when it comes to wind power prediction, especially long-term forecasting, the Bi-LSTM model can be considered one of the prevalent and precise methods owing to its ability to memorize data [11].

Another fundamental aspect of robustness in power system scheduling is the consideration of uncertainty. Although recent DNN-aided methods can extend predictions with guarantees, the occurrence of outliers is always expected. Accordingly, for the sake of comprehensiveness, it is essential to consider likely variations through prevalent methods. Conventionally, the Monte Carlo Simulation (MCS) has been utilized in order to overcome uncertainty by exploiting the probability distribution function. Despite its implementation in some cases, it lacks any relations for a model, rendering it perform completely randomly. Some DNN-based methods such as the Variational Autoencoders (VAE) and Generative Adversarial Network (GAN) models can be taken into account as alternatives. The GAN models, as unsupervised generative deep models, excel in generating realistic scenarios and data, which are valuable for making informed decisions and optimizing power flow. The main idea of the GAN model is a competition between two conflicting components, wherein the generator and discriminator aim to minimize and maximize their objective functions, respectively. Although the conventional GAN model is practicable in most cases, there is a possibility of mode collapse and struggle with convergence. Accordingly, a more reliable and stable variant, named WGAN, is introduced by Ref. [12] that resolves the convergence issue. In Ref. [12], the authors employed the WGAN model to generate wind power scenarios based on the historical data which results in a high correlation between generated and actual data. Known as Wasserstein GAN, to generate diverse samples. Consequently, the GAN model can be considered to generate adequate datasets required for the learning phase.

One of the fundamental aspects of power system optimization is to determine the optimal schedules which can be carried out by some well-known programs including Unit Commitment (UC) and Optimal Power Flow. While achieving the optimal schedules in a power system can be quite challenging, a well-designed platform can streamline the process. In this way, the Python for Power System Analysis module (PyPSA), a Python-based energy management toolbox, stands as a qualified alternative due to its flexibility and adaptability when compared to conventional optimal power flow methods [13]. Recently, the DNNs have been utilized to solve various kinds of OPF problems whether DC or AC [14]. In line with [15], an AC-OPF prediction model is constructed with the purpose of predicting the voltage of all buses, which is then used to compute the final optimization in an efficient way. In the previous study, it was proven that the proposed deep-based voltage-constrained model could excel in terms of accuracy and speed when compared to shallow learning-based models. The utilization of deep-based models in the field of optimal scheduling prediction is not limited to the AC power flow. Considering deep-based structures to cope with optimal scheduling in the power system is beneficial in terms of runtime and scalability. As another deep-based model, the Convolutional Neural Network (CNN) has a widespread application in various fields. Authors in Ref. [16] addressed a CNN-based method to carry out the AC-OPF with a high speed compared to the traditional approaches. Furthermore, a CNN architecture is used in Ref. [17] to solve the UC problem under the security constraints for a power system by means of minimizing the BESS operating cost. Although in previous research the proposed model could succeed in reducing the operating costs and computational time, it considered only 200 specific scenarios.

Although many studies considered DNN-based methods to achieve optimal scheduling in a power system, most of them suffer from a lack of a comprehensive and hierarchical framework that incorporates time-series prediction for a time-dependent variable like wind power as well as power system optimization. In fact, each deep architecture is particularly well-suited for specific types of modeling. For instance, despite the high ability of convolutional layers in extracting features, they cannot solely perform well in dealing with time-series variables such as wind power. On the other hand, the memorization ability of recurrent-based methods, like Bi-LSTM, makes them appropriate for time-series modeling. According to the proposed model in Ref. [18], a one-dimensional convolutional structure is used along with an LSTM model to predict the ratio of charging stations for the new energy vehicles to the charging stations of conventional vehicles. In a more complicated architecture, the study [19] modified the prediction performance by using wide first-layer kernels (WDCNN) in conjunction with Bi-LSTM. Regarding the mentioned reference, the cooperative

combination achieves higher accuracy, which may not be feasible for individual applications. Further, in Ref. [20], the CNN-BiLSTM structure is developed to predict the possible fluctuation on the demand side and determine the optimal schedules for large-scale distribution generators through a multi-objective model. In the previous study, it was shown that the hybrid design resulted in the highest score in comparison with the case utilizing only Bi-LSTM or CNN. Meanwhile, the mentioned research investigates the impact of an efficient algorithm to adjust initial weights on the accuracy improvement.

Concerning to above explanations, it is clear that the existing studies that are relevant to this topic have not considered at least one of the following:

- Utilizing three advanced ML methods —Random Forest (RF), Decision Tree (DT), and K-nearest Neighbors (KNN)— for recovering a significant volume of missing wind power observations across various time windows.
- An inclusive exploration of hyperparameters optimization for each CNN and BiLSTM model which are tuned by the OPTUNA, and evaluating its choices.
- Imperative performance of the proposed Optimal Scheduling Prediction (OSP) in predicting optimal schedules for the power system and wind farms.

The perpetual need for optimal scheduling in power systems, driven by both technical and financial considerations, underscores the importance of developing comprehensive frameworks capable of determining the optimal scheduling for days ahead. While significant progress has been made in accurately predicting wind power, there remains a critical gap in addressing the uncertainty inherent with a novel approach. Conventionally, scenario generation is a solution to this issue using methods based on distributed probability concepts. However, these approaches often generate scenarios that are entirely random and may not sufficiently capture the underlying patterns and dynamics of wind power generation. In contrast, this study introduces a novel approach by leveraging GANs for scenario generation. In our framework, the GAN-based model generates scenarios for wind power that closely align with historical predictions from the preceding 10 days. In other words, the proposed GAN model can estimate the possible fluctuation of wind power based on the previous 10 days' observations. This ensures that the generated scenarios are not only realistic but also representative of the underlying dynamics of wind power generation, thus enhancing the accuracy of the optimization process. This scenario generation approach can significantly augment the predictions related to the optimal dispatch of wind power. Furthermore, the hierarchical nature of the proposed framework leads to achieving optimality in each stage which is not possible through a single-stage procedure.

As a result, it is necessary to develop a comprehensive optimization model that establishes an accurate framework for predicting wind power and then these forecasted results will serve as input for the GAN model to generate valid and large enough synthetic scenarios. In other words, the GAN model is proposed as a qualified alternative to conventional probable distribution methods to generate different scenarios. Afterwards, these generated scenarios are used in power system optimization which is carried out by the PyPSA and Particle Swarm Optimization (PSO). The outcomes from this process are then utilized to build a comprehensive deep-based model that can predict the optimal scheduling. Based on the stated points, this paper proposes a comprehensive method that operates on the following contributions:

1 **Efficient Data Acquisition:** To ensure the dataset integrity and accuracy of the final prediction model, three ML methods, including Random Forest (RF), Decision Tree (DT), and K-nearest Neighbors (KNN), are used to address missing data in the wind power observations.
2 **OPTUNA-aided Bi-LSTM Architecture:** Recognizing the criticality of precise wind power prediction, the initial parameters such as learning rate, activation function, and batch size are optimized by OPTUNA for enhancing performance.
3 **Wind Power Fluctuation Modeling:** A CNN-based GAN model is designed to generate wind power fluctuation by considering historical data from the preceding ten days, facilitating more accurate predictions.
4 **Two-stage Optimization:** To reach the highest optimality, the power system optimization is implemented throughout two-stage planning, including PyPSA which is specified for the MIP programming, and the PSO which is suitable for the NLP programming.
5 **OSP model:** Introducing two deep learning-based models, namely CNN and Bi-LSTM, in order to swiftly and accurately predict optimal scheduling for the power system and optimal dispatch of wind power, thereby facilitating informed decision-making processes.

Also, the paper is organized as follows: Section 2 introduces basic theories utilized in the process of modeling. Section 3 applies the proposed deep-based models and carries out power system optimization. Finally, Section 4 reveals the conclusion.

2. Methodology

This section provides a comprehensive overview of the proposed model's framework. Initially, the overall framework is elucidated in detail in subsection 2.1. Following this, detailed explanations of the primary mathematical formulations concerning preprocessing and wind power prediction are provided in subsections 2.2 and 2.3, respectively. Subsequently, attention is directed toward the main deep-based models pertaining to wind and power system analysis, delineated in subsections 2.4 to 2.7. Finally, the proposed hyperparameter optimizer utilized in this study is demonstrated in 2.8.

2.1. Overall framework

In this part, the overall procedure of the proposed model is given. The first stage of the proposed model is designed to implement the data recovery phase. Accordingly, the data acquisition is completed by three different machine-learning methods. This stage is necessary in terms of dataset preparation for the wind power prediction stage. Meanwhile, the coefficient of determination is considered to select the best method to fill the missing data. Then the wind power will be predicted for the next 10 days. To execute this task, a fine-tuned Bi-LSTM model, which is achieved by the OPTUNA optimizer, is implemented with the aim of accurate wind power prediction. When it comes to accurate and reliable optimization, an adequate dataset is necessary to train the model. So, in the next stage, the GAN model is used to generate 1000 synthetic scenarios with high accuracy. The generated scenarios are given as input to the power system optimization model, which is involved by the PyPSA and PSO modules.

Firstly, the PyPSA module optimizes the generator status considering a UC framework. A binary variable, denoted as $C^{d,t}$, is used to represent the d^{th} generator status at the time t. Executing this stage enables us to minimize the operating costs of generators, and simultaneously achieve the maximum acceptable wind power because the solver tends to use the full potential of wind power. Following this, the upper limit of wind power is reached through this stage. The PyPSA module [21], as an open-source Python-based environment for power system analysis, outperforms other power system optimizers in terms of computational runtime and is compatible with other Python-based modules. In the second step, the generator status is fixed by considering $C^{d,t}$ as a parameter, while it tends to strive to optimize the power of generators. Since the format of this stage contains non-linear terms, it intends to be optimized by the PSO. Additionally, the output power of wind farms was

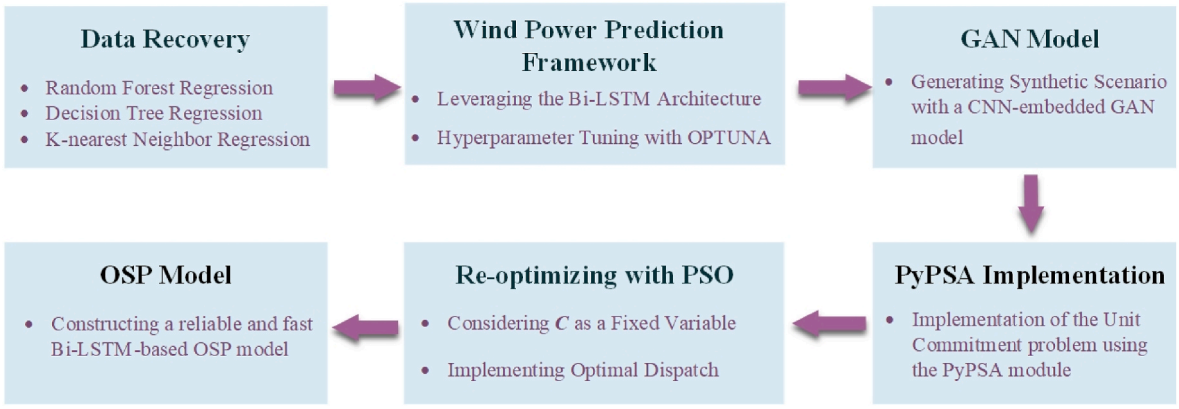

Fig. 1. Framework of the proposed model.

limited to the upper level, obtained from the first stage. This ensures both minimum cost and optimal schedules for wind farms. After fulfilling all stages, the outcomes are utilized to build a comprehensive prediction model that can receive the input of the network and compute the best optimal schedules as fast and accurately as possible. The overall framework proposed in this study is observable from Fig. 1.

2.2. Data sets

In this article, the collected data related to Ref. [22] is used to predict wind power. Four pivotal features have been selected to create a comprehensive repository for model training. These features are sin *a*, cos *b*, actual wind power of the control area, and actual wind power. The reason for selecting these features is their strong correlations among others. The higher the correlation provided during the feature selection stage, the more effective the training process becomes. Also, "*a*" signifies the number of days in a year, while "*b*" quantifies the duration of each day, measured in hours [23]. The operational time analysis spans from 31/12/2014, at 23:15:00, and extends its reach until 30/9/2020, at 23:00:00. Each timestep is delineated by a 15-min interval, resulting in a dataset containing 201,600 observations.

2.3. Data recovery

There is a possibility of missing some points of the dataset due to metering devices' malfunctions. Also, these unknown data should be filled in because training a model with incomplete data leads to a less practical and accurate model. Hence, at this stage, data recovery is accomplished using three prevalent ML-based regressors. Recent studies have explored the potential of ML methods, such as K-nearest Neighbors (KNN), to address the missing data issue [24]. In this regard, three regressors have been selected to complete the data acquisition task, including the KNN, Decision Tree (DT), and Random Forest Regression (RFR).

2.3.1. K-nearest neighbors

Firstly, the KNN is utilized to fit the dataset and discern underlying hidden relationships among wind observations. Based on the prior illustrations, the KNN, denoted as Unsupervised Kernel Regression (UKR), fills the gaps based on the average value of its nearest neighbors. In other words, the main idea of KNN is based on the assumption that if the close neighbors of x in the local space are labeled as y, it is highly possible that x behaves similarly and belongs to the y class. This method can be computationally expensive when confronted with a large dataset, but its strength lies in its simplicity of implementation. As formulated in Eq. (1), this method calculates the mean of values for the function $F(x')$, given a set of k-nearest neighbors as inputs.

$$F_{KNN}(x') = \frac{1}{K}\sum_{\theta \in N_{K(x')}} y_\theta \tag{1}$$

Where θ represents the index of each neighbor for the input of x'. The number of neighbors (K) should be tuned for the best result.

2.3.2. Decision tree regressor

Another used regressor is the DT which has been examined to complete the data recovery task. The DT regressor recursively splits data into subsets which can be appropriate for understanding the non-linear relationships [25]. This method is more robust to outliers and more interpretable when compared to RFR. However, it requires proper pruning to prevent overfitting.

$$\bar{y}_m = \frac{1}{n_m}\sum_{y \in Q_m} y \tag{2}$$

$$H(Q_m) = \frac{1}{n_m}\sum_{y \in Q_m} (y - \bar{y}_m)^2 \tag{3}$$

Where y indicates the label vector and $\in \mathbb{R}^l$. Moreover, the dataset at node m is denoted by Q_m, containing n_m samples. The prevailing criterion for the DT regression is MSE employed as the loss function, represented by $H(\bullet)$, which requires minimization. In fact, the objective function aims to find predicted values that closely resemble the learned mean values ($\bar{y}_m$).

2.3.3. Random Forest regressor

Finally, the RFR, as an ensemble learning method, has been employed to execute data retrieval, leveraging multiple DT regressors to increase its accuracy. In contrast with individual methods like KNN, the ensemble methods are offered to deal with large-scale missing data due to their rapid training time. Also, scalability and robustness to noise in ensemble methods such as RFR are significantly higher than KNN or DT [26]. Consequently, it is suitable for models with high-dimensional and complex datasets. The RFR process begins by randomly sampling observations and employing binary splitting in the first tree, iteratively reaching terminal nodes. Subsequently, the cost function is determined by averaging across all trees, facilitating an ensemble prediction approach for regression tasks. In order to evaluate its performance, the following generalization error, shown in Eq. (4), can be computed.

$$PE^* = P_{X,Y}[mg(X,Y) < 0] \tag{4}$$

Where $mg(X,Y)$ is the margin function, defined in Eq. (4). Also, X and Y are the distribution of random vectors [27].

$$mg(X,Y) = av_k I[h_k(X,\Theta_k) = Y] - \max_{j \neq Y} av_k I[h_k(X,\Theta_k) = j] \tag{5}$$

In Eq. (5), Θ_k represents independent identically distributed random vectors in which $k = \{1,2,3,...,K\}$. Moreover, $av(\bullet)$ and $I[\bullet]$ are average and indicator, respectively.

2.4. Wind power prediction

In this stage, the Bi-LSTM model is used to predict the wind power. Unlike the LSTM architecture, Bi-LSTM can train its parameters in both forward and backward directions simultaneously. This enhances the accuracy of the feature extraction process. In this structure, the purpose of the backward layer is to calculate the derivative of propagated errors of the forward layer [28]. Eqs. (6)–(11) express the Bi-LSTM formulation with N_l layers.

$$i_l^t = sigmoid\left(U_l^t h_l^{t-1} + W_l^t x_l^t + \theta_{li}\right) \tag{6}$$

$$f_l^t = sigmoid\left(U_l^t h_l^{t-1} + W_l^t x_l^t + \theta_{lf}\right) \tag{7}$$

$$O_l^t = sigmoid\left(U_l^t h_l^{t-1} + W_l^t x_l^t + \theta_{lo}\right) \tag{8}$$

$$\widetilde{C}_l^t = tanh\left(U_l^t h_l^{t-1} + W_l^t x_l^t + \theta_{lc}\right) \tag{9}$$

$$C_l^t = \left(f_l^t \otimes C_l^{t-1}\right) + \left(i_l^t \otimes \widetilde{C}_l^t\right) \tag{10}$$

$$h_l^t = O_l^t \otimes \left(tanh\left(C_l^t\right)\right) \tag{11}$$

Where W_l^t and x_l^t denote the weight matrix and input information, respectively. Additionally, i_l^t is the input of the lth layer at t. Also, $l = 1, 2, \ldots, N_l$. Through Eq. (7), the forget gate (f_l^t) will decide whether the information of the previous cell state should be forgotten or not. In Eqs. (9) and (10), the $\widetilde{C}_l^t$ and C_l^t are the present and candidate cell states, respectively. The $\widetilde{C}_l^t$ must be updated at each time. Also, the C_l^{t-1} is delineated as long-term memory. To achieve the outcome of Bi-LSTM's output gate, denoted by h_l^t, output information (O_l^t) and cell state (C_l^t) have been used. The θ_{li}^t, θ_{lf}^t, and θ_{lo}^t are the bias vectors associated with the input, forget, and output gates.

2.5. Evaluation criteria

In order to assess the prediction performance and its accuracy, a set of evaluation metrics has been employed. Considering the model's characteristics and the nature of the prediction task, emphasis is placed on regression metrics in this study. For this purpose, some formulas for evaluation indicators are defined in Eqs. (12)–(16).

$$MAE = \frac{1}{N}\sum_{j=1}^{N}\left|y_{j,target} - y_{j,pred}\right| \tag{12}$$

$$NMAE = \frac{100}{N}\sum_{j=1}^{N}\left|\frac{y_{j,target} - y_{j,pred}}{y_{j,target}^{max}}\right| \tag{13}$$

$$MSE = \frac{1}{N}\sum_{j=1}^{N}\left(y_{j,target} - y_{j,pred}\right)^2 \tag{14}$$

$$RMSE = \sqrt{MSE} \tag{15}$$

$$NRMSE = \frac{100}{y_{j,target}^{max}}\sqrt{MSE} \tag{16}$$

Where $y_{j,pred}$ and $y_{j,target}$ are the predicted and actual values relating to the j^{th} dataset. Also, $\bar{y}_{target}$ denotes the average of the actual vector. Eqs. (12) and (13) indicate Mean Absolute Error (MAE) and Normalized MAE (NMAE), respectively. Both MAE and NMAE are appropriate for those cases in which the impact of outliers can be ignored. On the other hand, Mean Squared Error (MSE), formulated in Eq. (14), is highly recommended when the outliers play a vital role in the optimization model. Hence, it seems that those metrics that are less dependent on rarely-happened disturbances are eligible to evaluate a time-varying variable like wind power. Furthermore, Root MSE (RMSE) and Normalized Root MSE (NRMSE), shown in Eqs. (15) and (16), are utilized as two frequently used criteria to assess the prediction model. In addition to the mentioned evaluation metrics, in some cases such as measuring the scores of regressors, calculation of the score is needed. In Eq. (17), the coefficient of determination (R-squared) formula is demonstrated.

$$R^2 = \frac{\sum_{j=1}^{N}\left(y_{j,pred} - \bar{y}_{target}\right)^2}{\sum_{j=1}^{N}\left(y_{j,target} - \bar{y}_{target}\right)^2} \tag{17}$$

Additionally, the normalization of the wind power dataset in the preprocessing step is carried out by the Min-Max Scalar from the scikit-learn library [29], as presented in Eq. (18).

$$x_{i,scaled} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{18}$$

2.6. The proposed Generative Adversarial Network

In this paper, a GAN model has been employed to generate a sufficiently large dataset for predicting wind power. Its setup comprises two deep-based modules: the generator and the discriminator, which act as adversaries. The generator's task is to produce new samples based on a random noise vector, while the discriminator's task is to determine whether the produced samples are real or not. In essence, the generator is trained under unsupervised conditions, while the discriminator is constructed as a supervised classifier [30]. The definitions of the generator and discriminator of the GAN model are as follows:

$$\min_G \max_D V(D,G) = E_{x\sim P_r}\, log(D(x)) + E_{z\sim P_{noise}}\, log(1 - D(G(z))) \tag{19}$$

Where $P_r(x)$ represents the probability distribution function of the real dataset, z is the random variable, and P_{noise} denotes the latent space. $G(z)$ represents the sample produced by the generator. A successful GAN model is achieved when the generator can effectively deceive the discriminator. In other words, the generator should continuously reduce its losses during the fitness process. It is important to note that in the GAN model, the generator's structure should be sufficiently efficient and complex to compete constructively with the discriminator.

2.7. Power system optimization

In this section, the power system framework is introduced. In many studies, the concept of multi-stage optimization is considered to deal with complexity and provide a concentration on a particular objective function in each stage [31]. In this paper, a two-stage optimization is considered in order to determine the upper and lower levels of optimization. The lower level is achieved by the implementation of the first stage, which is the UC problem using the PyPSA module. Here, the primary aim is to minimize the total cost associated with the ON/OFF status of generation units, yielding optimal schedules. Subsequently, the upper optimization level is determined by the second stage which is the optimal power flow solved by the PSO module. This consecutive framework can first optimize the generation units' states and secondly optimize their dispatched power.

2.7.1. PyPSA implementation

According to the two-stage optimization, the PyPSA module is employed in the first stage to perform UCs for all generators. In other words, the PyPSA is responsible for solving the Mixed-integer Pro-

gramming (MIP), and determining the optimal status of each generator unit. Therefore, the 1000 scenarios provided by the GAN model along with the 1000 different scenarios for demand produced by uniform distribution are given to PyPSA in order to accomplish the first stage of power system optimization. The objective function of the proposed optimization model is formulated in Eq. (20). As it is shown in this equation, the PyPSA aims to optimize the binary variable $C^{d,t}$, representing the state of dth generator at the time t.

$$F\left(C^{d,t}, P_G^{d,t}, P_{Wind}^{w,t}, P_{LS}^{l,t}\right) = min\left(S_{susd} + S_{fuel} + S_{loadshedding}\right) \tag{20}$$

In the provided equation, the symbol S indicates the cost value, measured in $. Eq. (20) determines the main objective function in the PyPSA model. This objective function (F) considers four variables: commitment ($C^{d,t}$), generator power output ($P_G^{d,t}$), wind farm output ($P_{Wind}^{w,t}$), and load shedding value ($P_{LS}^{\gamma,t}$). Accordingly, S_{susd} and S_{fuel} represent the total start-up/shutdown and fuel costs, respectively. Eq. (21) indicates start-up and shutdown costs, and Eq. (22) represents the total fuel cost. Also, $S_{loadshedding}$ corresponds to the penalty cost associated with the load shedding which is computed via Eq. (23).

$$S_{susd} = \sum_{d=1}^{D}\sum_{t=1}^{T} S_{startup}^{d,t} + S_{shutdown}^{d,t} \tag{21}$$

$$S_{fuel} = \sum_{d=1}^{D}\sum_{t=1}^{T} M^d P_G^{d,t} + CP^d P_G^{d,nom} \tag{22}$$

$$S_{loadshedding} = \sum_{\gamma=1}^{L}\sum_{t=1}^{T} \lambda_{ls} P_{LS}^{\gamma,t} \tag{23}$$

In Eq. (22), the M^d and CP^d denote the marginal and capital prices of the d^{th} generator. Also, the load shedding cost is calculated by Eq. (23). The γ stands for the number of loads. The value of the load-shedding penalty coefficient is 1000 $\frac{\$}{MW}$. Based on the PyPSA structure, the startup and shutdown costs are considered constraints along with their related parameters. Constraints (24) and (25) demonstrate how S_{susd} and S_{fuel} are calculated.

$$S_{startup}^{d,t} \geq SUC^d\left(C^{d,t} - C^{d,t-1}\right) \forall d \in D, t \in T \tag{24}$$

$$S_{shutdown}^{d,t} \geq SUC^d\left(C^{d,t-1} - C^{d,t}\right) \forall d \in D, t \in T \tag{25}$$

With regard to the above constraints, SUC^d and SDC^d determine prices of start-up and shutdown for d^{th} generator based on $. The increase and decrease rate limits of generator production are defined in constraint (27). Also, Eqs. (28) and (29) are the minimum uptime and downtime constraints of generators. The generator production is restricted by upper and lower limits, defined in Eq. (30).

$$C^{d,t} \in \{0,1\} \forall d \in D, t \in T \tag{26}$$

$$RD^d \leq P_G^{d,t} - P_G^{d,t-1} \leq RU^d \quad \forall d \in D, t \in T \tag{27}$$

$$\sum_{\tau=t}^{t+MUT^d} C^{d,\tau} \geq MUT^d\left(C^{d,t} - C^{d,t-1}\right) \forall d \in D, t \in T \tag{28}$$

$$\sum_{\tau=t}^{t+MDT^d} \left(1 - C^{d,\tau}\right) \geq MDT^d\left(C^{d,t-1} C^{d,t}\right) \forall d \in D, t \in T \tag{29}$$

$$P_G^{d,min} C^{d,t} \leq P_G^{d,t} \leq P_G^{d,max} C^{d,t} \quad \forall d \in D, t \in T \tag{30}$$

2.7.2. PSO implementation

After determining the optimal status of generators and wind farms, the next step is to determine the optimal schedules, which is formulated as a Mixed-integer Non-linear Programming (MINLP) problem. Consequently, in the second stage of power system optimization, a PSO algorithm is employed to tackle the non-linearity. Eqs. (31) and (32) represent the PSO formulation for p particles [32].

Table 1
PSO parameters.

No.	Parameter	Value
1.	Swarm population	55
2.	Cognitive coefficient (c_1)	1.54
3.	Social coefficient (c_2)	1.98
4.	Damping factor (w_{damp})	0.998
5.	Maximum iterations (It_{max})	35
6.	Constriction factors (phi_1 and phi_2)	2.05, 2.05

$$v_p^{t+1} = \omega v_p^t + c_1 r_1\left(PB_p - x_p^t\right) + c_2 r_2\left(GB - x_p^t\right) \tag{31}$$

$$x_p^{t+1} = x_p^t + v_p^{t+1} \tag{32}$$

Where v_p^{t+1} and x_p^{t+1} embody the velocity and position of pth particle at time $t+1$, respectively. Alongside these fundamental entities, ω, c_1, and c_2 are inertia weight, cognitive, and social coefficients used in the PSO model. Random variables r_1 and r_2, both ranging between 0 and 1, shape the trajectory of each particle. In Eq. (31), PB_p is the best local position found by particle p so far, and GB is the best global position attained by any particle within the swarm. The objective function of the PSO model is the quadratic form of the cost function. Moreover, the constriction factor is utilized in the PSO model to dynamically change the velocity for decreasing the search space [33]. More detailed information about the proposed PSO model is provided in Table 1.

2.8. Hyperparameter optimizer

Due to the indispensable stance of hyperparameter tuning in DNN-based models to achieve meticulous results, an adequate module must be considered in each DNN model. On this basis, some research, like [34], utilized modified heuristic methods to adjust the hyperparameters. Meanwhile, OPTUNA, an open-source optimization framework, outperforms others in terms of speed and accuracy. In other words, its advanced pruning mechanism can effectively reduce the time of finding the optimal solutions so that it can be a better choice for hyperparameter regulation [35]. In the literature [36], the Gray Wolf optimizer has been used to set the kernel function parameters in a short-term wind power prediction framework. Furtherly, in Refs. [37,38], the genetic and artificial fish swarm algorithms have been employed as the hyperparameter regulators to enhance the prediction accuracy. As another example, the authors in Ref. [39] present an adaptive and Bayesian-optimized Bi-LSTM model for selecting the best configuration. In this paper, to optimize the deep model hyperparameters, the OPTUNA library is used. OPTUNA is an automatic cost-effective optimizer. It uses various samplers such as random, Bayesian, and genetic calculations during the optimization process [40]. Based on Eq. (33), OPTUNA models the objective function as probabilistic surrogate models like Tree-structured Parzen Estimators (TPE) or Gaussian processes to seek the best configuration of hyperparameters.

$$\overset{*}{\theta} = \underset{\theta}{arg\ min}\Psi(\theta) \tag{33}$$

Where $\overset{*}{\theta}$ and θ are the set of hyperparameters and the search space. Also, $\Psi(\theta)$ is the objective function that has to be optimized by the selected hyperparameters [41].

2.9. Optimal scheduling prediction model

In the final stage, the proposed OSP model is trained to predict the optimal schedules for the next days. These optimal schedules are

Table 2
Setting up the OSP model.

Input (X)	Output (Y)
Generator Capacities	Optimal Power Dispatched by PSO
Upper Level of Wind Power	Optimized Wind Power by PSO
Load-shedding Capacity	Load-shedding Specification by PSO
Demand	Total Production Obtained from PSO
Input Shape: (1000, 59, 24)	Output Shape: (1000, 59, 24)

generators' power, wind power, curtailed wind power, and the amount of load shedding at each time step. The input and output shapes consist of 1000 matrices, each comprising 59 rows (representing the number of generators) and 24 columns (representing the time intervals). As it is evident, this highly efficient model outperforms traditional optimal power flow methods in terms of runtime.

During the training of this model, specific features were designated as inputs (X) and outputs (Y), illuminated in Table 2. In fact, the major aim of the OSP model is to learn how to map X to Y.

To train the last proposed deep model, a Bi-LSTM architecture consisting of three hidden layers is employed. The input sequence possesses a shape of (59, 24), while the output takes the form of a 24-dimensional vector. Throughout the training phase, a dataset comprising 990 scenarios for training and validation, reserving 10 scenarios (or days) for testing purposes, is utilized. This parameter can substantially optimize the runtime by hindering the learning process when overfitting occurs.

3. Results and discussions

In this section, all results related to the deep models and their hyperparameter values are provided during each stage. Also, to validate the effectiveness of the proposed model, the 118-bus network with 54 generators is considered in the power system optimization stage. All of the network parameters were obtained from the PYPOWER Case 118 [42].

Additionally, numerical experiments are conducted using Python language, version 3.11.3, on a 64-bit Windows computer equipped with an Intel® Core™ i7-12650H CPU and 16 GB of RAM. Furthermore, the following Python libraries were employed for simulation:

- Keras and TensorFlow: Deep learning implementation [43,44].
- Pandas and NumPy: Data analysis [45,46].
- OPTUNA: Hyperparameter tuning [47].
- PyPSA: Power system analysis [21].
- Matplotlib: Output visualization [48].

3.1. Data recovery results

Fig. 2 demonstrates the results of data acquisition implemented by three regressors. Based on this figure, three periods have been considered in which missing data occurs. The first time interval, during which relevant data is absent, spans from 49823 to 50208 timesteps, corresponding to the period from May 29, 2016, at 22:45:00 to June 2, 2016, at 22:45:00. Similarly, the second and third missing time intervals occur from December 7, 2016, at 01:45:00 to December 12, 2016, at 02:45:00, and from October 24, 2016, at 01:45:00 to October 31, 2016, at 22:45:00, respectively.

Some parameters related to the KNN method such as number of neighbors, weight function, and leaf size are considered 3, uniform, and 30 respectively. Also, the maximum depth of each decision tree for DT and RF regressors is set to 100.

Meanwhile, evaluation is conducted using the R^2 score metric as a criterion. Analysis from Table 3 reveals that the DT regressor outperforms in fitness process, yielding the highest score among the methods considered. Thus, the entire gaps were filled by the DT regressor. Furthermore, in terms of computational efficiency, the DT regressor demonstrates reasonable performance, whereas the RFR regressor can be computationally expensive. Despite the relatively shorter runtime observed with the KNN regressor, it is crucial to consider its lower accuracy, which outweighs its advantage in runtime efficiency.

3.2. Wind power prediction results

The results associated with the Bi-LSTM architecture with the attention layer mechanism that has been used to accurately predict wind power in the next 10 days are provided in the following. Also, the results of three other prevalent alternatives are provided for comparison with the proposed model. Before training the proposed wind power prediction model, it is necessary to normalize and split data sets in order to enhance the model's efficiency. Once the preprocessing is completed and the input data sets are established, a deep-based structure can be designed by means of predicting wind power for 10 days ahead. This structure includes two Bi-LSTM layers, two attention layers, two time-

Table 3
Scores of proposed regressors are used to estimate missing data.

Regressor	KNN	DT	RFR
Score	0.9046	0.9999	0.9748
Runtime (seconds)	0.3502	28.3235	171.7220

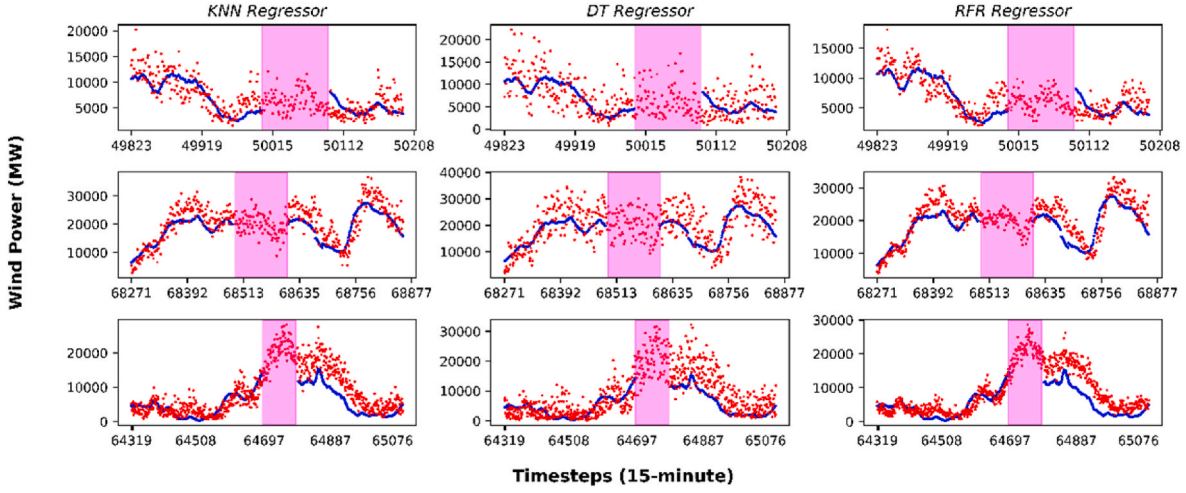

Fig. 2. Comparison of KNN, DT, and RFR methods in terms of executing the data recovery phase.

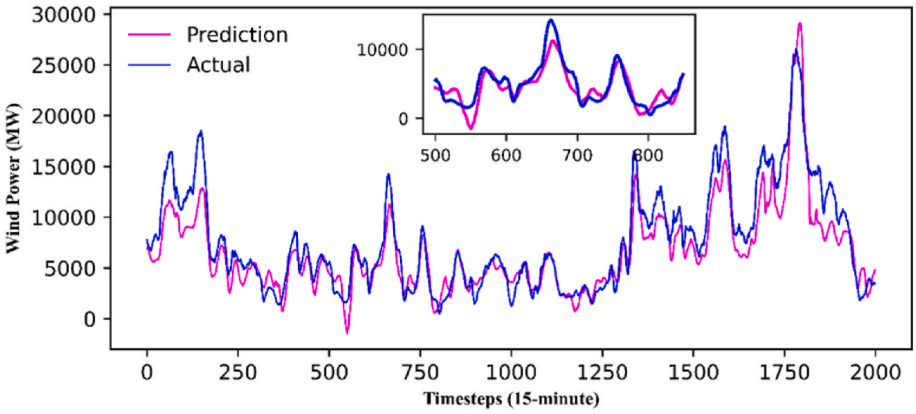

Fig. 3. Wind power prediction through Bi-LSTM structure for 2000 time intervals.

Table 4
Evaluation of wind power prediction model for training and testing sets.

Approach	Evaluation (MSE)	
	Training Datasets	*Test Datasets*
Bi-LSTM	0.0820	0.0819
LSTM	0.0831	0.0917
GRU	0.0963	0.1057
RNN	0.2006	0.2082

distributed wrappers for attention layers, one layer for flattening, and one dense layer as output.

In the training phase, the total shape of the dataset related to the wind power prediction stage is (200160, 3, 960), where 60 % of the data is allocated to the training dataset and 20 % to the validation dataset. Based on the tensor shape, the first term refers to the samples used to train the proposed model. The second term indicates the number of features considered in the training phase. The last term, valued at 960, indicates the look-back parameter. In simpler terms, the model is configured to forecast wind power for ten days ahead. Given 15-min time intervals (where each 96-intervals represents one day), the model must learn from 960 timesteps in each iteration.

Fig. 3 shows the prediction of wind power executed by the proposed Bi-LSTM model along with actual values. The evaluation performances of each structure used to predict wind power are gathered in Table 4. As it is clear, the Bi-LSTM structure records the lowest error. Additionally, the importance of the bidirectional layer can be concluded from this table, as the LSTM model demonstrates higher error rates compared to the proposed model.

In Fig. 4, the correlation plots for utilized structures are depicted. Additionally, the Pearson correlation is calculated for each approach, where the proposed model can achieve the highest correlation.

The number of trials considered for the OPTUNA optimization is 20. Also, the 16th attempt has the best performance and the corresponding value of the objective function is 0.0239. The objective function is defined based on the loss value of the validation data set. Furthermore, the patience parameter is defined for early stopping to prevent overfitting. Some hyperparameters optimized by the OPTUNA have been shown in Table 5.

The $\left\{\overrightarrow{\sigma}_1, \overrightarrow{\sigma}_2\right\}$ and $\left\{\overleftarrow{\sigma}_1, \overleftarrow{\sigma}_2\right\}$ are the forward and backward activation functions of the first and second Bi-LSTM structures, respectively.

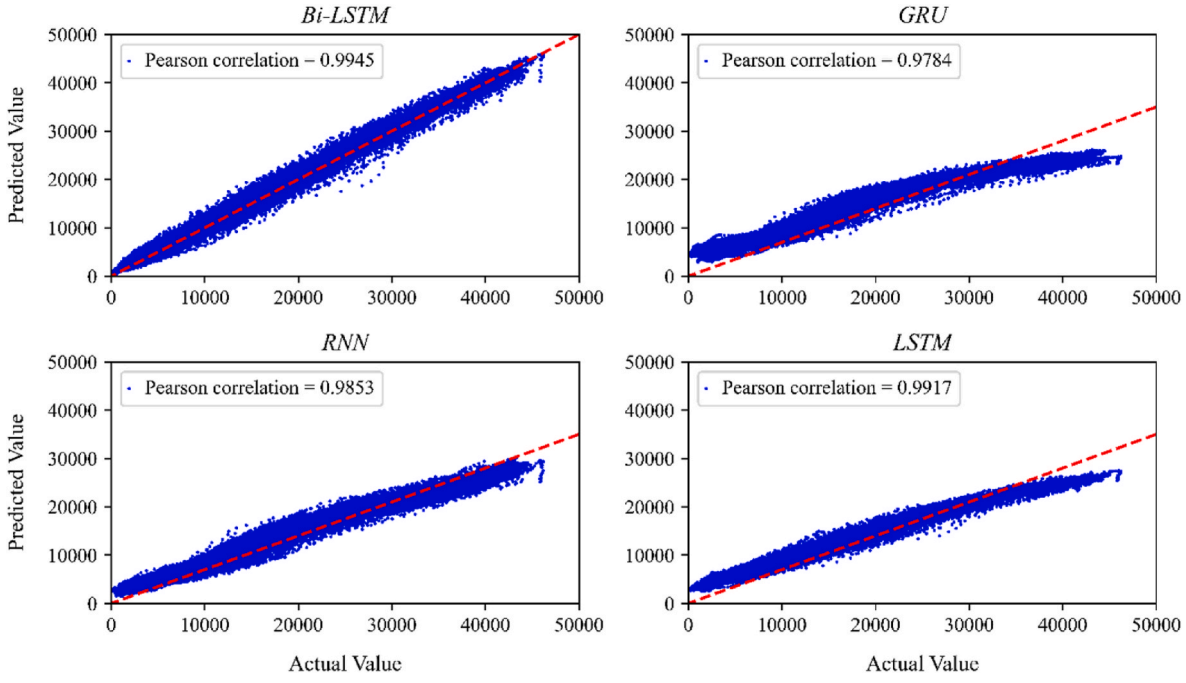

Fig. 4. Correlation comparisons among four proposed architectures used for the wind power prediction.

Table 5
Optimized hyperparameters by OPTUNA in the wind power prediction stage for the proposed model.

Hyperparameters		Decision Space	Optimal Value
Bi-LSTM$_1$	$\overrightarrow{\sigma}_1, \overleftarrow{\sigma}_1$	[ReLU, Tanh, Linear]	[Tanh, ReLU]
	Neurons	[10, 500]	146
Bi-LSTM$_2$	$\overrightarrow{\sigma}_2, \overleftarrow{\sigma}_2$	[ReLU, Tanh, Linear]	[ReLU, ReLU]
	Neurons	[10, 500]	146
Dense Layer	Units	[1, 100]	28
	L_1	[1e-5, 1e-2]	0.004341
	L_2	[1e-5, 1e-2]	0.005204
Set Up the Model	Learning Rate	[1e-8, 1e-2]	0.00011172
	β_1	[0, 1)	0.632841
	β_2	[0, 1)	0.316708
	ε	[1e-8, 1e-4]	1.101947e-06
	Batch Size	[32, 256]	118
	Clipnorm	[0.1, 1]	0.594708
	Clipvalue	[0, 1]	0.529121

The loss function used in the wind power prediction stage is MAE. For two LSTM layers of the wind power prediction model, three common activation functions are considered in which the best one will be determined by the OPTUNA optimizer. Also, the search space for LSTM layers' units is defined from 10 to 500. Besides, some influential hyperparameters are considered in the training phase which is controlled by the OPTUNA achieving more effective fitness.

3.3. Scenario generation

The results associated with the proposed GAN model are provided in this section. In line with the above explanations, the proposed GAN model is responsible for generating synthetic data for predicted wind power as similar as possible to the original predicted values. To construct the proposed GAN model, one-dimensional convolution layers along with dense layers are used in both generator and discriminator structures. Convolutional layers are highly suitable when the complexity of the model necessitates the extraction of intricate information. Following that, Fig. 5 shows the fitness process of the GAN model.

Based on the above figure, in the beginning, the generator is not substantially successful in deceiving the discriminator. However, after a number of epochs, it starts learning properly how to produce fake values with the highest reality. It can be seen that the generator reaches the loss value of 0.5329, after executing 1000 epochs. In addition, it is crucial for both the generator and discriminator to converge, ensuring the validity of the model. Following this, the discriminator converges with a loss value of 0.7933. The learning rates for the generator and discriminator are set to 5e-5 and 1e-5, respectively. Additionally, the selected batch size for training the GAN model is 6.

Concerning the mentioned GAN model, some major information related to the GAN model's configuration is brought in Table 6. Based on this table, the discriminator receives an array with the shape of 24 from the generator. If this array meets the accuracy criteria, then the discriminator decides to accept it. Conversely, if generated values deviate significantly from real data, the discriminator must refuse it. Therefore, the discriminator acts as a binary filter, and its output shape will be a single value. Additionally, some policies are considered to refrain from overfitting, such as applying Batch Normalization and Dropout layers.

3.4. Results of the power system optimization

In the power system optimization stage, three wind farms located at buses 30, 57, and 77 are considered. The nominal capacity of wind farms is set to 10,000 MW. Moreover, the generator cost coefficients and upper/lower limits are taken from http://motor.ece.iit.edu/Data/118bus_ro.xls [49].

Fig. 6 represents the commitments for one scenario executed by PyPSA. The red points indicate that the corresponding generator is ON. This step ensures the lowest operational cost for all generators. Depending on each dataset of wind power, the result of the UC problem can vary.

After solving the MIP problems with the PyPSA module, the proposed PSO model determines the optimal dispatches in the second stage. In this way, Fig. 7 shows the changes in PSO's cost functions for ten sample scenarios. In most cases, the proposed PSO model could be converged after 35 iterations, achieving optimal values for corresponding decision

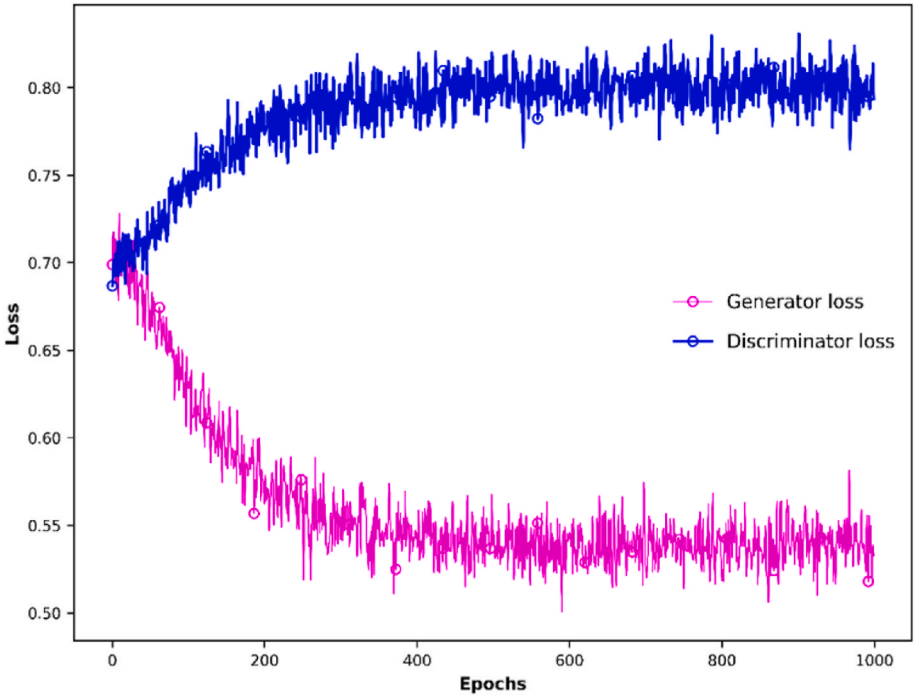

Fig. 5. Generator and discriminator loss curves during the training process of the GAN model.

Table 6
Considered parameters for the GAN model.

Discriminator	**Input Layer:** One-dimension convolutional layer; Activation Function: ReLU; Units: 24 Flatten Layer Dense Layer; Units: 128 Batch Normalization 10 % Dropout Dense Layer; Units: 512 Batch Normalization 10 % Dropout **Output Layer:** Dense Layer; Activation Function: Sigmoid; Units: 1
Generator	**Input Layer:** One-dimension convolutional layer; Activation Function: Units: 24 One-dimension convolutional layer; Units: 128 Flatten Layer Dense Layer; Units: 512 Dense Layer; Units: 128 Dense Layer; Units: 32 **Output Layer:** Dense Layer; Activation Function: Sigmoid; Units: 24

variables.

3.5. Results of the optimal scheduling prediction model

In this section, the analysis results of the final stage of optimization are provided. To implement the proposed model, two deep approaches are used along with some prevalent metrics in order to choose the best approach. Both methodologies undergo tuning via the OPTUNA optimizer, while early-stopping techniques are applied to mitigate the risk of overfitting. Patience value in the early-stopping techniques has been set to 3 which implies if the validation loss fails to decrease for three consecutive iterations during the training process, it will trigger an automatic halt to the learning process. The OPTUNA selected arranges for both mentioned approaches are provided in Table 7. According to the OPTUNA model, 50 trials are considered for the optimization process in which the best objective value, 0.001218, belongs to the 22nd iteration.

The results of fitness procedures associated with the Bi-LSTM and Convolution-based structures are depicted in Fig. 8. For both approaches, the changes of loss values related to the test and validation data sets through a hundred iterations are provided.

With regard to the above figure, the performance of Bi-LSTM architecture has proved more robust, reaching a lower error when compared to its convolution-based counterpart. On the other hand, the convolutional alternative displays erratic behavior via a hundred iterations.

Furthermore, the percentile indicator is used to evaluate the accuracy of the results obtained by the prediction model. Specifically, the 0.05 and 0.95 percentiles are defined as the lower and upper limits of the permissible range. The 0.95-percentile represents the value that 95 % of the entire data is lower than. Also, only 5 % of the entire results are lower than the 0.05-percentile. Following that, Table 8 illustrates what percentage of predicted values are in range and what percentage of them are out of range. This table illuminates how accurate the proposed model is in predicting optimal schedules for ten days ahead.

Fig. 9 is prepared to compare the actual demand and the predicted production by the prediction model at each timestep for 10 days ahead. According to Table 8, the majority of the values are in the permissible range. As an example, the 4th scenario falls entirely within the acceptable range, a fact that is evident in Fig. 9 for day 4.

Aside from determining the optimal schedules, the proposed model can be significantly practical in the prediction of optimal wind power dispatch. As can be seen in Fig. 10, the trained OSP model can reasonably forecast dispatched wind power for ten days ahead. The corresponding tensor shape for both the actual and prediction outcomes is (10, 24). It should be mentioned that two main factors could substantially decrease the accuracy of the proposed model, including the length of sequence and ensemble nature.

In Fig. 11, the actual and predicted wind power are compared sequentially. Notably, the sustained accuracy of predictions over a span of ten days underscores the robustness and reliability of the proposed model.

Additionally, based on Fig. 12, the Pearson correlation coefficient for the aforementioned prediction stands at 92 %, which demonstrates the practicality of the ten-day prediction framework.

Based on Table 9, the execution time of the power system optimization using the conventional method (PyPSA and PSO utilizing) is compared with the proposed model. To implement the conventional

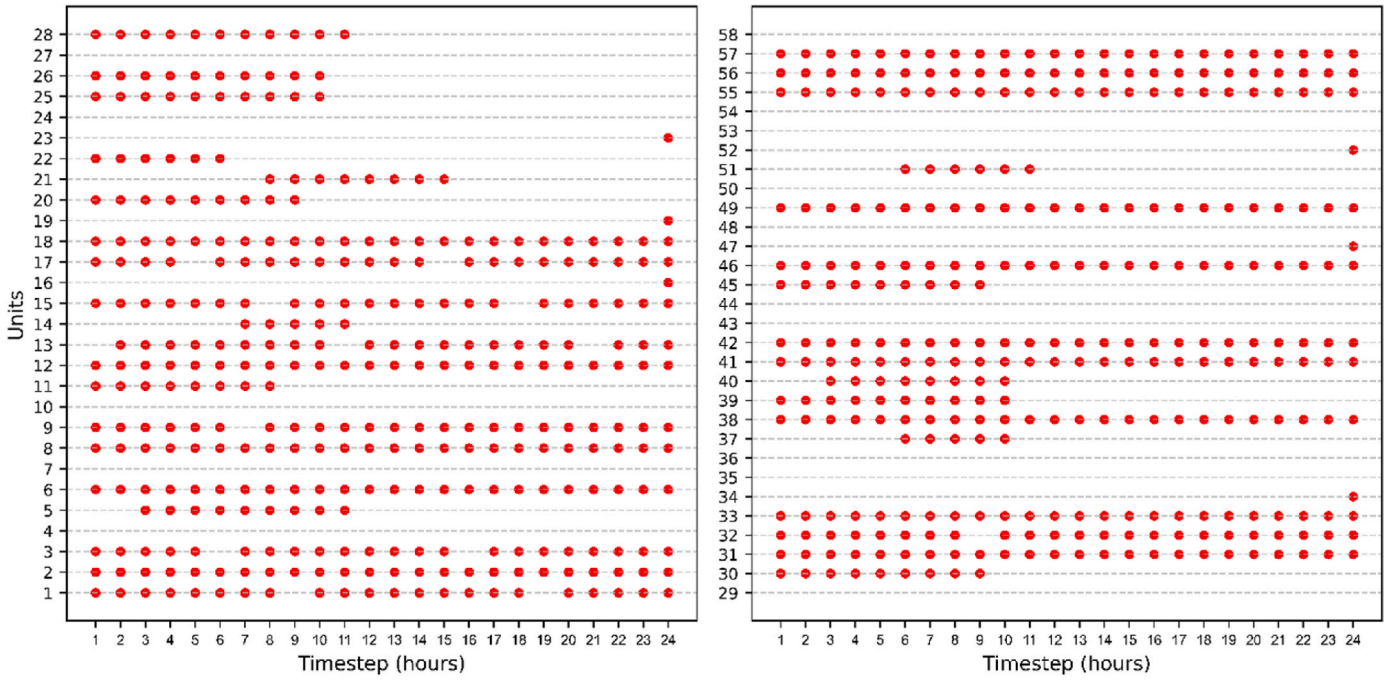

Fig. 6. Result of the UC problem solved by the PyPSA module for one scenario.

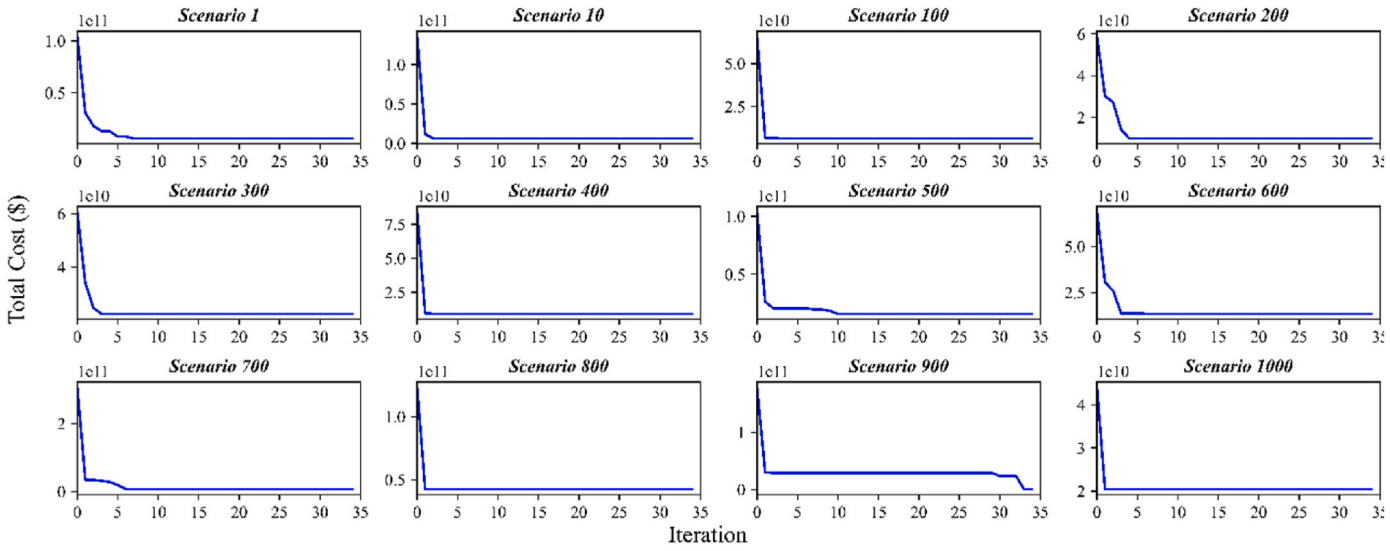

Fig. 7. The convergence process of 12 scenarios optimized by PSO.

Table 7
Optimized hyperparameters by OPTUNA in the last stage.

Hyperparameters		Decision Space	Optimal Value
Bi-LSTM	$\overrightarrow{\sigma}, \overleftarrow{\sigma}$	[ReLU, Tanh, Linear]	[Tanh, ReLU]
	Neurons	[32, 512]	104
Dense 1	Units	[10, 256]	220
Dense 2	Units	[10, 256]	225
Dense 3	Units	[10, 256]	48
Optimizer	Category	[Adam, RMSprop]	[Adam]
Set Up the Model	Learning Rate	[1e-8, 1e-2]	0.003247
	β_1	[0, 1)	0.563419
	β_2	[0, 1)	0.658289
	ε	[1e-8, 1e-4]	4.088532e-05
	Batch Size	[2, 256]	2
	Clipnorm	[0.1, 1]	0.751259
	Clipvalue	[0, 1]	0.335522

optimization with a multi-stage structure, both PyPSA and PSO models must be used consecutively. The total runtime of the two declared models exceeds 7 h for 1000 different scenarios. The proposed model optimized the entire scenario datasets in just 0.3829 s.

In Table 10, the computational runtime of the proposed model along

Table 8
Evaluation performance of 10 days prediction based on the upper and lower percentiles.

Scenarios	In Range (%)	Out of Range (%)
Day 1	87.50	12.50
Day 2	95.83	4.16
Day 3	87.50	12.50
Day 4	100.00	0.00
Day 5	91.66	8.33
Day 6	87.50	12.50
Day 7	83.33	16.66
Day 8	87.50	12.50
Day 9	91.66	8.33
Day 10	87.50	12.50

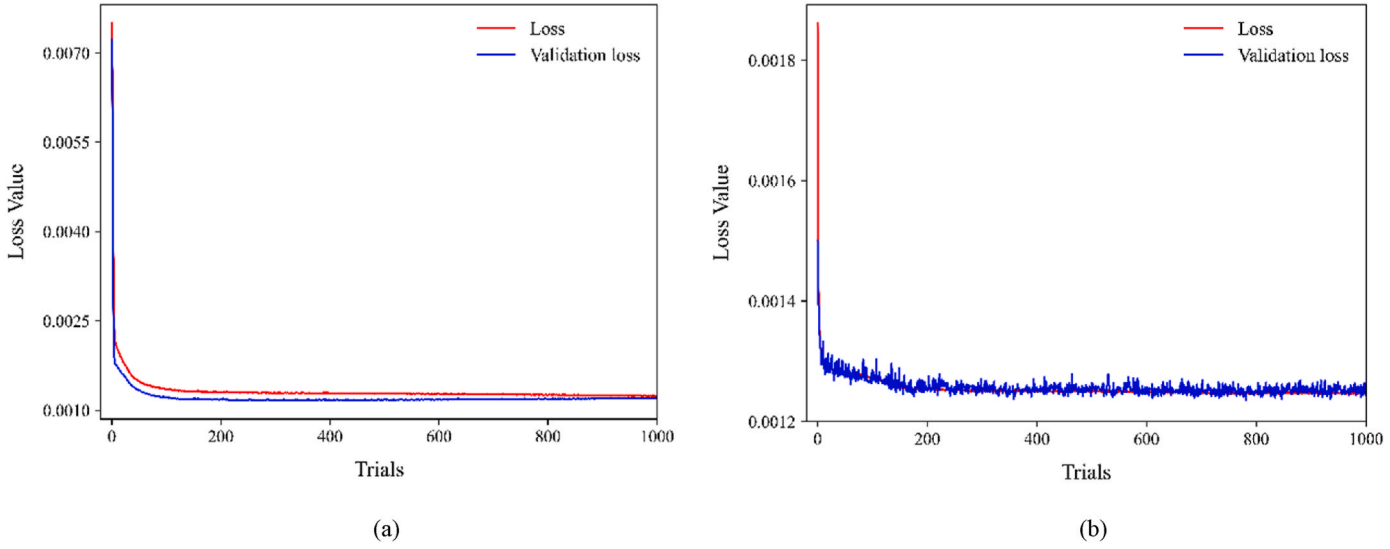

Fig. 8. The convergence process of two proposed approaches via a hundred trials: (a) fitness process of Bi-LSTM approach. (b) Fitness process of convolution-based approach.

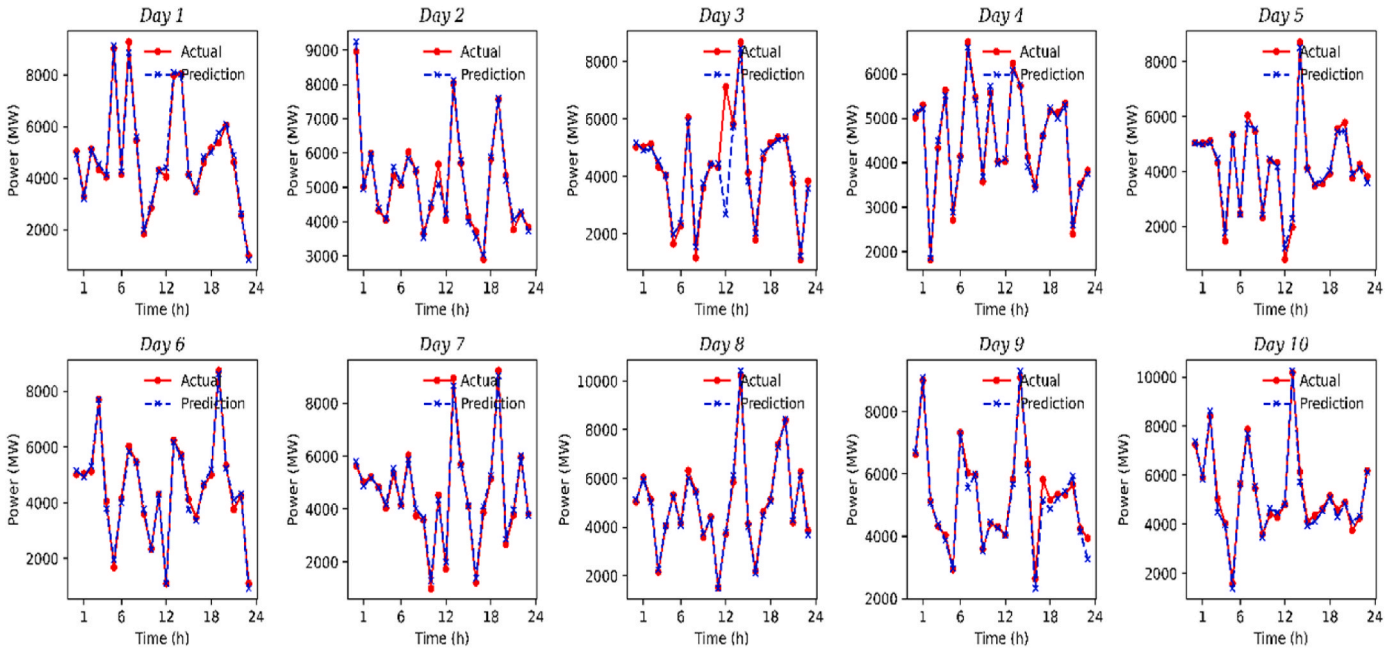

Fig. 9. Comparison of actual demand and total dispatched power obtained from last stage prediction.

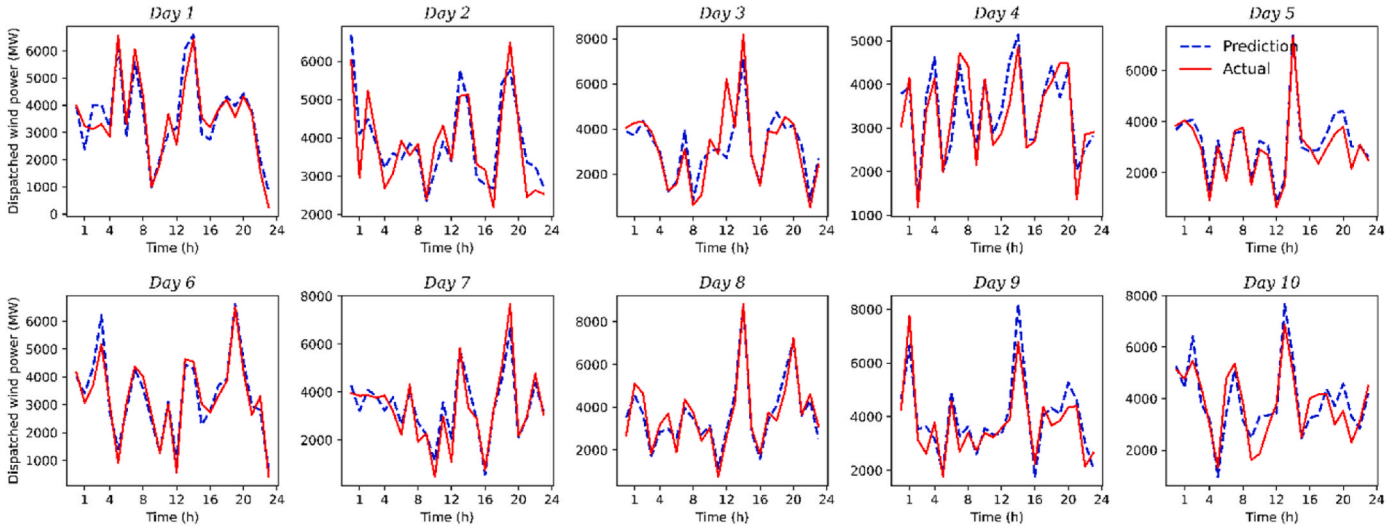

Fig. 10. Comparison of prediction and actual values of dispatched wind power for each day.

with other models is illustrated. It should be considered that the runtime of the proposed model pertains to the prediction of one hundred scenarios. This consideration underscores the efficiency and scalability of our approach in handling the complexities inherent in forecasting multiple scenarios, further enhancing its practical utility in real-world applications.

3.6. Sensitivity analysis

To establish comparative conditions and evaluate the hyperparameter values selected by OPTUNA, a sensitivity analysis has been conducted in this section. As indicated in Table 11, when the OSP model is reconstructed with a 1-layer convolution, the model's performance accuracy decreases. All of the hyperparameters are the same as the Bi-LSTM structure. According to the mentioned table, some of the loss evaluators have been brought.

Accordingly, the performance of the OSP model can deteriorate by about 8 % if the convolutional layer is used instead of the proposed Bi-LSTM structure. When the activation function of the forward Bi-LSTM layer is changed from Tanh to ReLU, the results in Table 12 are obtained. Also, Table 13 evaluates the MSE metric by altering the optimizers. Performance evaluations have been given for both training and testing datasets. As shown in Table 7, OPTUNA selects the Adam optimizer which yields a lower MSE value compared to the RMSprop optimizer. These results ensure the integrity of the OPTUNA optimizer.

Regarding the concluded results, utilizing the most prevalent optimizer can diminish the amount of error by around 24 % for the test data set. Also, the performance of the proposed model can be enhanced by about 3.5 % in the case of using the Tanh activation function instead of ReLU in the first layer of the Bi-LSTM model.

4. Conclusions

This paper proposed a multi-faceted framework designed to develop

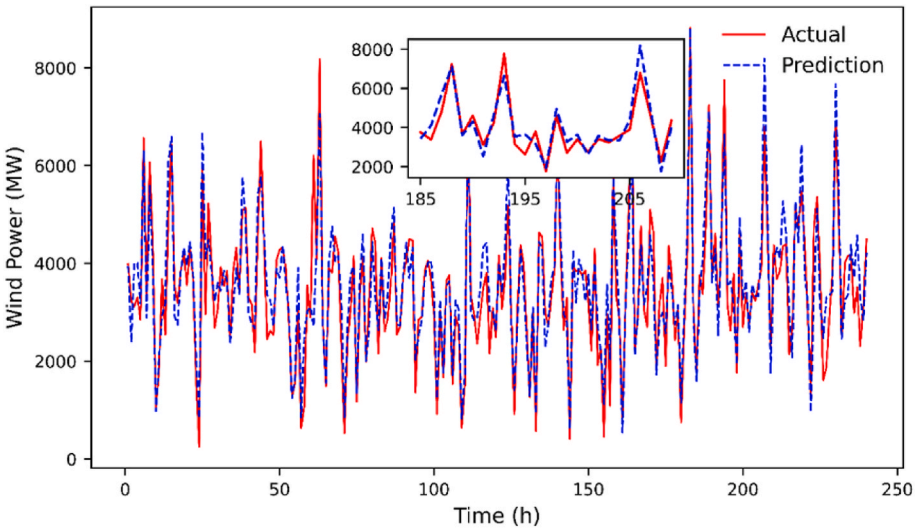

Fig. 11. Comparison of wind power prediction and actual wind power relating to three wind farms for 10 days ahead.

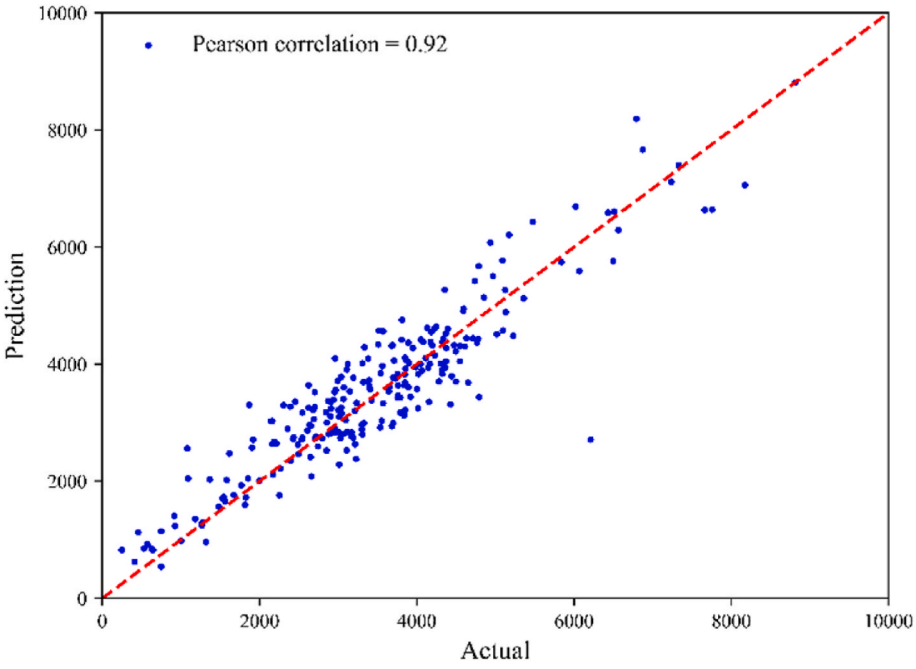

Fig. 12. Correlations between actual and predicted values of dispatched wind power.

Table 9
Comparison of the run-time of PyPSA and PSO implementation with the proposed model.

Approach	PyPSA	PSO	Proposed Model
Time (s)	8702.88	19257.70	0.3829

Table 10
Comparison between the proposed model performances with other related models.

Model	Network	Number of samples (train and test)	Considered scenarios	Running time
Proposed	118-bus	200,160	1000	0.3829 *s*
[16]	30-bus	8000	1	0.101 *ms*
[50]	300-bus	60,000	1	1.70 *ms*
[51]	118-bus	60,000	1	0.29 *ms*

a robust deep-based optimal scheduling prediction model. In the initial stage, the data acquisition task is executed by three ML methods and a comparison is made based on their performances. After choosing the DT regressor as the best regressor, the wind power prediction is accomplished by an OPTUNA-aided Bi-LSTM architecture to deliver remarkably accurate wind power predictions up to 10 days in advance. The achieved outcomes reveal a commendable error rate of 0.08 when compared to other prevalent methods.

For the sake of scarcity of wind power, a CNN-based GAN model is employed to generate sufficient scenarios for wind power as if the generated scenarios are closely similar to the predicted values. Subsequently, the predicted results transform into the GAN model, yielding a hundred synthetic and precise wind power scenarios. Both components of the GAN model, discriminator, and generator are optimized by the

Table 11
Performance comparisons of the Bi-LSTM model as the main and CNN-based model.

Evaluation	Main Model	Convolution-based Model
MSE	0.00119108	0.00129133
RMSE	0.03451212	0.03593519
NRMSE (%)	3.46743225	3.61040828
MAE	0.01024018	0.01173786
NMAE (%)	1.02883162	1.17930311
R^2	0.84453342	0.82979040

Table 12
Sensitivity analysis results: Determining the impact of activation function on the forward Bi-LSTM layer.

$\vec{\sigma}$	Tanh		ReLU	
Evaluation	Train	Test	Train	Test
MSE	0.00114580	0.00119108	0.00117318	0.00123322

Table 13
Sensitivity analysis results: Determining the impact of optimizers.

Optimizer	Adam		RMSprop	
Evaluation	Train	Test	Train	Test
MSE	0.00114580	0.00119108	0.00131528	0.00147788

OPTUNA, reaching the acceptable error of 0.8 and 0.5, respectively. Once these GAN-generated data align closely with real data, they are seamlessly integrated into the next stage which is the power system optimization. The GAN-generated scenarios transition to the PyPSA module, where they are utilized to execute the UC problem to determine the optimal schedules for the considered 118-bus network. Then, the PSO model re-optimizes all scenarios, incorporating non-linearity terms.

Finally, in the ultimate stage, a Bi-LSTM prediction model tuned by the OPTUNA optimizer is constructed to predict the optimal scheduling for ten days ahead. The results obtained from the proposed model demonstrate the decrease of runtime by 100 times compared to the conventional method, from almost 466 min to lower than 0.4 s. Moreover, in the last prediction stage and from the sensitivity analysis results, the proposed Bi-LSTM design proves more practical compared to the convolutional counterpart. Other findings from the sensitivity analysis section underscore the validity of OPTUNA choices in selecting the best adjustments. In future endeavors, a multi-objective function structure into our multi-faceted framework can be incorporated which can contribute to the optimal decision-making process.

CRediT authorship contribution statement

Ali Peivand: Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation. **Ehsan Azad Farsani:** Writing – review & editing, Validation, Supervision, Project administration, Methodology. **Hamid Reza Abdolmohammadi:** Validation, Software, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. Li, H. Jiang, P. Yang, P. Fan, P. Qi, L. Kang, Small signal stability analysis and optimize control of large-scale wind power collection system, IEEE Access 10 (2022) 28842–28852.

[2] D.K.Y. Zong, A. Thavlov, O. Gehrke, H.W. Bindner, Application of model predictive control for active load management in a distributed power system with high wind penetration, in: IEEE Transactions on Smart Grid vol. 3, June 2012, pp. 1055–1062, https://doi.org/10.1109/TSG.2011.2177282, 2.

[3] M.A. Kousounadis-Knousen, I.K. Bazionis, D. Soudris, F. Catthoor, P.S. Georgilakis, A new Co-optimized hybrid model based on multi-objective optimization for probabilistic wind power forecasting in a spatiotemporal framework, IEEE Access (2023).

[4] J. Dixon, W. Bukhsh, C. Edmunds, K. Bell, Scheduling electric vehicle charging to minimise carbon emissions and wind curtailment, Renew. Energy 161 (2020) 1072–1091.

[5] G. López, P. Arboleya, Short-term wind speed forecasting over complex terrain using linear regression models and multivariable LSTM and NARX networks in the Andes Mountains, Ecuador, Renew. Energy 183 (2022) 351–368.

[6] D.-E. Choe, H.-C. Kim, M.-H. Kim, Sequence-based modeling of deep learning with LSTM and GRU networks for structural damage detection of floating offshore wind turbine blades, Renew. Energy 174 (2021) 218–235.

[7] Z. Niu, Z. Yu, W. Tang, Q. Wu, M. Reformat, Wind power forecasting using attention-based gated recurrent unit network, Energy 196 (2020) 117081.

[8] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.

[9] Y. Sun, X. Wang, J. Yang, Modified particle swarm optimization with attention-based LSTM for wind power prediction, Energies 15 (12) (2022) 4334 [Online]. Available: https://www.mdpi.com/1996-1073/15/12/4334.

[10] M.-S. Ko, K. Lee, J.-K. Kim, C.W. Hong, Z.Y. Dong, K. Hur, Deep concatenated residual network with bidirectional LSTM for one-hour-ahead wind power forecasting, IEEE Trans. Sustain. Energy 12 (2) (2020) 1321–1335.

[11] H. Chen, H. Liu, X. Chu, Q. Liu, D. Xue, Anomaly detection and critical SCADA parameters identification for wind turbines based on LSTM-AE neural network, Renew. Energy 172 (2021) 829–840.

[12] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks. Presented at the Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2017 [Online]. Available: https://proceedings.mlr.press/v70/arjovsky17a.html.

[13] P. Colbertaldo, F. Parolin, S. Campanari, A comprehensive multi-node multi-vector multi-sector modelling framework to investigate integrated energy systems and assess decarbonisation needs, Energy Convers. Manag. 291 (2023) 117168.

[14] X. Pan, M. Chen, T. Zhao, S.H. Low, DeepOPF: a feasibility-optimized deep neural network approach for AC optimal power flow problems, IEEE Syst. J. 17 (1) (2022) 673–683.

[15] W. Huang, X. Pan, M. Chen, S.H. Low, Deepopf-v: solving ac-opf problems efficiently, IEEE Trans. Power Syst. 37 (1) (2021) 800–803.

[16] Y. Jia, X. Bai, L. Zheng, Z. Weng, Y. Li, ConvOPF-DOP: a data-driven method for solving AC-OPF based on CNN considering different operation patterns, IEEE Trans. Power Syst. 38 (1) (2022) 853–860.

[17] T. Wu, Y.-J.A. Zhang, S. Wang, Deep learning to optimize: security-constrained unit commitment with uncertain wind power generation and BESSs, IEEE Trans. Sustain. Energy 13 (1) (2021) 231–240.

[18] B. Li, X. Lv, J. Chen, Demand and supply gap analysis of Chinese new energy vehicle charging infrastructure: based on CNN-LSTM prediction model, Renew. Energy 220 (2024) 119618.

[19] D. Geng, B. Wang, Q. Gao, A hybrid photovoltaic/wind power prediction model based on Time2Vec, WDCNN and BiLSTM, Energy Convers. Manag. 291 (2023/09/01/2023) 117342, https://doi.org/10.1016/j.enconman.2023.117342.

[20] K. Liu, et al., An energy optimal schedule method for distribution network considering the access of distributed generation and energy storage, IET Generation, Transmission & Distribution (2023).

[21] T. Brown, J. Hörsch, D. Schlachtberger, PyPSA: Python for Power System Analysis, 2017 arXiv preprint arXiv:1707.09913.

[22] Open Power System Data, doi: https://doi.org/10.25832/time_series/2020-10-06.

[23] E. Memmel, T. Steens, S. Schlüters, R. Völker, F. Schuldt, K. Von Maydell, Predicting renewable curtailment in distribution grids using neural networks, IEEE Access 11 (2023) 20319–20336.

[24] M. Jena, S. Dehuri, An integrated novel framework for coping missing values imputation and classification, IEEE Access 10 (2022) 69373–69387, https://doi.org/10.1109/ACCESS.2022.3187412.

[25] M.S.A.S. Mogos, X. Liang, C.Y. Chung, An effective very short-term wind speed prediction approach using multiple regression models, 3, in: IEEE Canadian Journal of Electrical and Computer Engineering vol 45, 2022, pp. 242–253, https://doi.org/10.1109/ICJECE.2022.3152524. Summer.

[26] K.F. Sotiropoulou, A.P. Vavatsikos, P.N. Botsaris, A hybrid AHP-PROMETHEE II onshore wind farms multicriteria suitability analysis using kNN and SVM regression models in northeastern Greece, Renew. Energy (2023) 119795.

[27] B.L.-M. Learn, in: Random Forest, vol 45, 2001 [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=+L.+Breiman%2C+Random+forests%2C+Mach.+Learn.%2C+vol.+45%2C+no.+1%2C+pp.+532%2C++2001.&btnG=.

[28] K.L.M.-S. Ko, J.-K. Kim, C.W. Hong, Z.Y. Dong, K. Hur, Deep concatenated residual network with bidirectional LSTM for one-hour-ahead wind power forecasting, in: IEEE Transactions on Sustainable Energy vol 12, April 2021, pp. 1321–1335, https://doi.org/10.1109/TSTE.2020.3043884, 2.

[29] O. Kramer, O. Kramer, Scikit-learn, Machine learning for evolution strategies (2016) 45–53.

[30] E. Cramer, L.R. Gorjao, A. Mitsos, B. Schäfer, D. Witthaut, M. Dahmen, Validation methods for energy time series scenarios from deep generative models, IEEE Access 10 (2022) 8194–8207.

[31] A. Peivand, E. Azad-Farsani, H.R. Abdolmohammadi, Wind curtailment mitigation in presence of battery energy storage and electric vehicle: a comprehensive multi-objective decision-support framework, J. Clean. Prod. 412 (2023) 137215.
[32] X. Song, et al., Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model, J. Petrol. Sci. Eng. 186 (2020) 106682.
[33] J.F. Schutte, A.A. Groenwold, A study of global optimization using particle swarms, J. Global Optim. 31 (2005) 93–108.
[34] J. Wang, Y. Qian, L. Zhang, K. Wang, H. Zhang, A novel wind power forecasting system integrating time series refining, nonlinear multi-objective optimized deep learning and linear error correction, Energy Convers. Manag. 299 (2024) 117818.
[35] S. Hanifi, S. Lotfian, H. Zare-Behtash, A. Cammarano, Offshore wind power forecasting—a new hyperparameter optimisation algorithm for deep learning models, Energies 15 (19) (2022) 6919.
[36] P. Lu, et al., A novel spatio-temporal wind power forecasting framework based on multi-output support vector machine and optimization strategy, J. Clean. Prod. 254 (2020) 119993.
[37] X. Wang, C. Wang, Q. Li, Short-term wind power prediction using GA-ELM, Open Electr. Electron. Eng. J. 11 (1) (2017).
[38] X. Zhai, L. Ma, Medium and long-term wind power prediction based on artificial fish swarm algorithm combined with extreme learning machine, International Core Journal of Engineering 5 (10) (2019) 265–272.
[39] N.D.M. Kaselimi, A. Voulodimos, E. Protopapadakis, A. Doulamis, Context aware energy disaggregation using adaptive bidirectional LSTM models, in: IEEE Transactions on Smart Grid vol 11, July 2020, pp. 3054–3067, https://doi.org/10.1109/TSG.2020.2974347, 4.
[40] S. Hanifi, A. Cammarono, H. Zare-Behtash, Advanced hyperparameter optimization of deep learning models for wind power prediction, Renew. Energy 221 (2024) 119700.
[41] M. Zhou, et al., Machine learning modeling and prediction of peanut protein content based on spectral images and stoichiometry, LWT 169 (2022) 114015.
[42] K. Paul, P. Sinha, S. Mobayen, F.F. El-Sousy, A. Fekih, A novel improved crow search algorithm to alleviate congestion in power system transmission lines, Energy Rep. 8 (2022) 11456–11465.
[43] F. Chollet, Keras: the Python Deep Learning Library, 2018.
[44] M. Abadi, et al., TensorFlow: large-scale machine learning on heterogeneous distributed systems, ArXiv (2016) abs/1603.04467.
[45] C.R. Harris, et al., Array programming with NumPy, Nature 585 (7825) (2020) 357–362.
[46] W. McKinney, P. Team, Pandas-Powerful python data analysis toolkit, in: Pandas—Powerful Python Data Analysis Toolkit, vol 1625, 2015.
[47] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: a next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2623–2631.
[48] E. Bisong, E. Bisong, Matplotlib and seaborn. Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners, 2019, pp. 151–165.
[49] H. Ye, Z. Li, Pricing the ramping reserve and capacity reserve in real time markets, arXiv preprint arXiv:1512.06050 (2015).
[50] W. Huang, X. Pan, M. Chen, S.H. Low, DeepOPF-V: solving AC-OPF problems efficiently, IEEE Trans. Power Syst. 37 (1) (2022) 800–803, https://doi.org/10.1109/TPWRS.2021.3114092.
[51] X. Pan, Deepopf: deep neural networks for optimal power flow. Presented at the in Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, 2021, November.