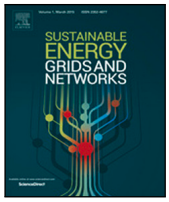
Sustainable Energy, Grids and Networks 40 (2024) 101525

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

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



Physics-informed convolutional neural network for microgrid economic dispatch

Xiaoyu Ge, Javad Khazaei *

Electrical & Computer Engineering Department, Lehigh University, 19 Memorial Drive, West Bethlehem, PA 18015, USA

ARTICLE INFO

Keywords:
Economic dispatch
Microgrid
Convolutional neural network
Physics-informed machine learning
Optimal dispatch

ABSTRACT

The variability of renewable energy generation and the unpredictability of electricity demand create a need for real-time economic dispatch (ED) of assets in microgrids. However, solving numerical optimization problems in real-time can be incredibly challenging. This study proposes using a convolutional neural network (CNN) based on deep learning to address these challenges. Compared to traditional methods, CNN is more efficient, delivers more dependable results, and has a shorter response time when dealing with uncertainties. While CNN has shown promising results, it does not extract explainable knowledge from the data. To address this limitation, a physics-inspired CNN model is developed by incorporating constraints of the ED problem into the CNN training to ensure that the model follows physical laws while fitting the data. The proposed method can significantly accelerate real-time economic dispatch of microgrids without compromising the accuracy of numerical optimization techniques. The effectiveness of the proposed data-driven approach for optimal allocation of microgrid resources in real-time is verified through a comprehensive comparison with conventional numerical optimization approaches.

1. Introduction

Microgrids offer an appealing option for addressing the difficulties posed by aging grid infrastructures and natural disasters on a local scale [1]. One of the key practical challenges in microgrid operation is economic dispatch (ED), which involves the optimal allocation of power generation dispatch to meet energy demand [2]. The task of finding the optimal microgrid operation can be viewed as an optimization problem that involves multiple soft and hard constraints. These constraints are linked to the specifications of power generation and the need to maintain a balance between energy production and demand, and can vary depending on the available energy resources and cost characteristics of the system [3–5].

1.1. Literature review

In microgrids, the power dispatch problem can be classified into different types based on varying objectives. In [6], an optimal energy dispatch algorithm is proposed. Another optimal energy dispatch method for an island energy hub with diverse sources is presented in [7]. Additionally, [8] introduces an event-triggered optimal dispatch algorithm for a multi-energy system.

Conventional approaches use a model-based numerical optimization techniques to solve the ED problem for microgrids [9–16]. For example, in [17], a stochastic optimization algorithm was proposed, which could efficiently solve the uncertainty of the renewable power generation in ED by implementing a new approximate dynamic programming method. Genetic algorithm is one of the most common methods for optimization problems, which was used in [18] to improve the convergence of ED problem and expedite the process of verifying the solutions' feasibility, efficiency, and quality. In addition, with the development of technology, the structure of loads also become diverse, many controllable loads, transferable loads, and heat loads are involved in ED problems. In [19], a multi-objective economic dispatch model considering electrical vehicles and transferable loads is established. Other examples for model-based ED problem include multi-objective optimization approach (theta-dominance-based evolutionary algorithm) with integrated decision-making (fuzzy c-means) [9], shuffled frog hopping algorithm with particle swarm optimization [11], frequency-constrained stochastic model to find the optimal generation schedule [13], or a mean-tracking model for stochastic economic dispatch [15].

In comparison to traditional model-based or numerical optimization-based approaches, machine learning-based methods are more efficient, robust, and less reliant on prior knowledge or resource cost characteristics. As the number of constraints increases, directly

* Corresponding author.
E-mail addresses: xig620@lehigh.edu (X. Ge), khazaei@Lehigh.edu (J. Khazaei).

https://doi.org/10.1016/j.segan.2024.101525
Received 16 June 2024; Received in revised form 17 August 2024; Accepted 9 September 2024
Available online 11 September 2024

solving an optimization problem can be computationally complex (e.g., 10–100 times slower than machine learning-based methods [20, 21]). Consequently, machine learning-based techniques have been suggested for solving economic dispatch problems in power systems [20–28]. For example, in [20], a random-forest regression model was proposed to predict the renewable power generation and local loads and find the optimization operation of a microgrid. In [23], a deep-recurrent neural network with long short-term memory (DRNN-LSTM) was proposed to forecast aggregated load and PV power output during a short-term period and find the optimal load dispatch. The proposed DRNN-LSTM model offered 7.43% improvement in forecasting and can reduce 8.97% of the daily cast. In [25], an artificial neural network (ANN) model was proposed to deal with the uncertainties brought by renewable generation via applying the nonlinear autoregressive exogenous model (ARIMA). In [26], the ED problem was modeled as a Markov decision process (MDP) and the authors proposed an improved advanced deep reinforcement learning algorithm to solve the ED problem in combined heat and power (CHP) system. In [27], the authors utilized the integration of artificial neural networks and Q-learning algorithms to resolve the optimal management problem of power grid operation and maintenance.

Previous studies have proposed various effective techniques for solving the economic dispatch problem in microgrids. However, these methods are not without limitations. Existing machine learning methods [20–29] heavily rely on collected data and often face difficulties due to operational condition changes. In addition, conventional machine learning techniques have limited generalization capabilities and their convergence rates are difficult to control. Additionally, other machine learning approaches such as reinforcement learning [27], decision trees and ARIMA [26] struggle to handle large datasets and require a significant training for accurate predictions, which is time consuming and inefficient. Furthermore, many existing studies only use machine learning methods for forecasting, rather than combining optimization and forecasting to improve performance. In [29], the authors proposed a CNN model to help solve the optimal flow problem and reduce the prediction running time significantly (350 X speed up). However, the proposed model is purely data-driven and does not include physical rules into the model. Finally, existing machine learning-based approaches cannot generally adapt to system changes/expansion and require a re-training if a component fails or new components are added. Physics-informed machine learning is an emerging direction in machine learning design that embeds physical laws during the training of a machine learning algorithm and has shown significant improvement in accuracy and adaptation of machine learning techniques to changing situations and uncertainties [30]. However, to our best knowledge, no existing study has explored physics-informed machine learning for economic dispatch problem of microgrids.

1.2. Contribution of this work

There is a clear knowledge gap on utilizing machine learning-based approaches for solving the economic dispatch problem in microgrids, which requires further investigation. This paper proposes a physics-inspired machine learning approach via convolutional neural networks (CNN) for solving the ED problem in real time. Due to the time-series nature of economic dispatch problem, two approaches including RNN and CNN could be used. However, CNN has shown promising results and outperformed RNN in many applications, such as data prediction, image classification, and video analysis [31], which has been selected in this work. Furthermore, CNN has multiple convolutional layers, which can improve its performance in extracting and learning features from data compared to deep neural networks (DNNs). The main contributions of this paper are listed as:

1. A 1D-CNN model was developed to solve the economic dispatch problem of a microgrid with renewable generation and conventional generation. Compared with existing deep learning-based approaches, the proposed CNN architecture is provides a more accurate prediction of solution of ED in real-time when intermittencies of renewable generation exists.
2. To solve the inability of existing data-driven techniques to account for physical constraints of the system, a physics-inspired CNN approach was developed to incorporate the constraints of the ED problem into the CNN design and increase the accuracy and efficiency of the proposed method.
3. Incorporating ED constraints in the training of the proposed CNN-based ED architecture significantly reduces the amount of training data required to predict ED solutions in real-time without compromising runtime.
4. Compared with a conventional ED approach based on numerical optimization, the proposed method is 400 times faster in converging to an optimal solution with an average accuracy more than 98%.

Several experiments showcase the accuracy, effectiveness, and real-time computational advantages of the proposed physics-inspired CNN-based ED compared with conventional numerical optimization or deep learning-based techniques.

The rest of the paper is organized as follows: Section 2 formulates the economic dispatch problem of microgrids and Section 3 includes the proposed machine-learning approach. Final results and analysis are included in Section 4 and Section 5 concludes the paper.

2. Economic dispatch of microgrids

Microgrid is a community-level power system typically composed of three components: generation units, energy storage units, and loads. The generation units generally encompass a range of resources, including conventional sources such as diesel generators (DS), natural gas generators (NG), and combined heat and power (CHP) systems, as well as renewable sources such as solar photovoltaic (PV) panels and wind turbines (WT). Battery is the primary component for energy storage, storing excess energy generated by renewable sources to serve as a backup power supply, thereby enhancing the stability of the microgrid. The load component consists of residential power consumption, commercial power consumption, and electric vehicles. However, in this study, only the generation units and the aggregated load (comprising residential power consumption, commercial power consumption, and electric vehicles) will be considered. Due to the varying costs and capacities of different generation units, as well as fluctuating load demands over time, providing a cost-effective yet stable power supply is a significant challenge in microgrid operation. Ensuring the cheapest power delivery while maintaining stability requires careful optimization and management. The economic dispatch (ED) problem is a typical optimization problem, intending to reduce system costs by rapidly shifting from higher to lower-cost generation before dispatching, all while ensuring a balanced load operation.

This problem for a set of diverse generations including CHP, NG, DS, PV, and WT, i.e., $G \in [CHP,\ NG,\ DS,\ PV,\ WT]$ can be formulated as a numerical optimization:

$$\min \sum_{t=1}^{T_i} \sum_{i=1}^{N_{G,i}} c_i(P_i^G(t)), \quad P_i^G \in G \tag{1a}$$

$$s.t. \quad \sum_{i=1}^{N_{G,i}} P_i^G(t) = P_{load}(t) \quad G \in \{CG, PV, WT\} \tag{1b}$$

$$U_{G,i}^t P_{G,i}^{\min} \le P_i^G(t) \le U_{G,i}(t) P_{G,i}^{\max}, \tag{1c}$$

$$P_i^G(t) - P_i^G(t-1) \le R_{\text{up}}^{\max} U_{G,i}(t-1) \tag{1d}$$

$$P_i^G(t-1) - P_i^G(t) \le R_{\text{down}}^{\max} U_{G,i}(t), \tag{1e}$$

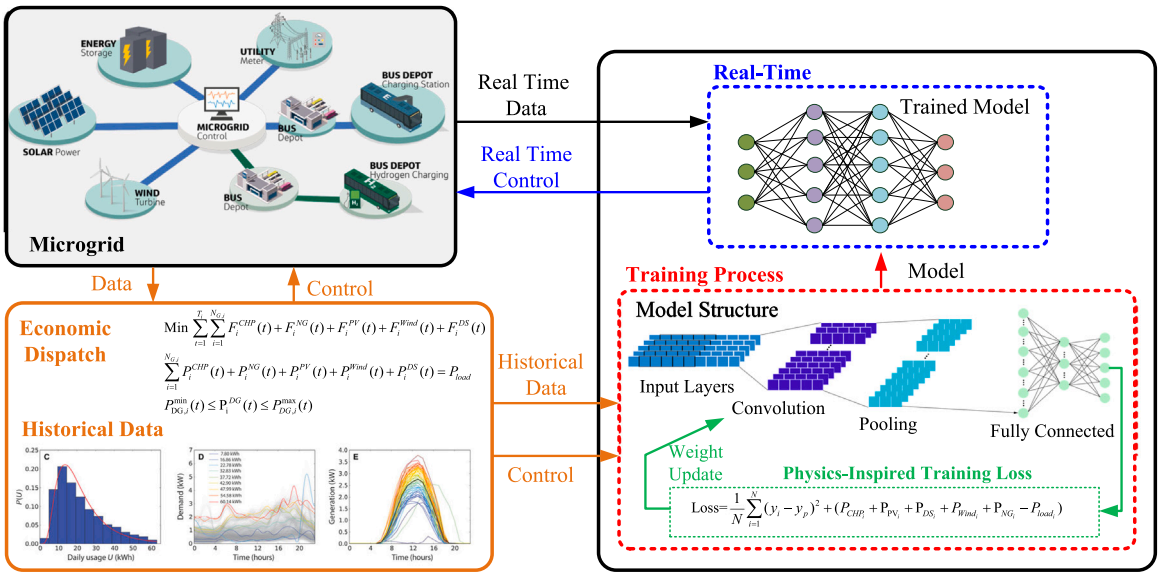


Fig. 1. Schematic of the proposed PI-CNN based microgrid economic dispatch.

$$P_{PV,i}^{min} \le P_i^{PV}(t) \le P_{PV,i}^{max} \tag{1f}$$

$$P_{WT,i}^{min} \le P_i^{WT}(t) \le P_{WT,i}^{max} \tag{1g}$$

where Eq. (1a) is the objective function of the optimization problem to minimize the overall generation cost of generation units c_i within over a time length T_i (minutes), where P_i^G (kW) is the generated power (dispatching command) of generation unit i. Constraints of this economic dispatch problem is formulated by (1b)–(1g), where the load balance is expressed by (1b). Constraints (1d) to (1e) represent the bounds of generation units and ramp rate conditions with $U_{G,i} \in \{0,1\}$ being the on/off status of the conventional generation, R_{up}^{max} and R_{down}^{max} in kW/min being the maximum ramp up and ramp down rates of conventional generators. The cost function c_i for conventional generation is a quadratic function of their generated power, which is formulated by:

$$c_i(P_i^{CHP}(t)) = \alpha_i^{CHP} + \beta_i^{CHP} + \gamma_i^{CHP}(P_i^{CHP}(t))^2 \tag{2}$$

$$c_i(P_i^{NG}(t)) = \alpha_i^{NG} + \beta_i^{NG} P_i^{NG}(t) + \gamma_i^{NG}(P_i^{NG}(t))^2 \tag{3}$$

$$c_i(P_i^{DS}(t)) = \alpha_i^{DS} + \beta_i^{DS} P_i^{DS}(t) + \gamma_i^{DS}(P_i^{DS}(t))^2 \tag{4}$$

where α, β, γ are the cost function coefficients of generation units. The output of renewable generation highly relies on the weather conditions, the source of the uncertainty. According to [32,33], and [34], the cost function of PV and WT are formulated as:

$$P_i^{PV}(t) = \rho_{stc}(1 + K_T(T_c - T_i)) \tag{5}$$

$$c_i(P_i^{PV}(t)) = K_i^{PV} P_i^{PV}(t) \tag{6}$$

$$P_i^{WT}(t) = \begin{cases} 0 & \text{if } v_t < v_{on} \text{ or } v_t > v_{off} \\ P_0 \dfrac{v_t - v_{on}}{v_0 - v_{on}} & \text{if } v_{on} \le v_t \le v_0 \\ P_0 & \text{if } v_0 \le v_t \le v_{off} \end{cases} \tag{7}$$

$$c_i(P_i^{WT}(t)) = K_i^{WT} P_i^{WT}(t) \tag{8}$$

Details of variables and their definition can be found in [34].

3. Physics-informed machine learning-based approach

The concept of physics-informed machine learning involves incorporating physical laws or prior knowledge into the learning process to enhance model performance. In this study, we introduce a physics-informed CNN (PI-CNN) model for solving the economic dispatch problem in microgrids. Additionally, we construct a DNN model to provide a comparison and demonstrate the superior performance of the PI-CNN. Fig. 1 depicts the structure of the proposed approach, where a classical ED problem is utilized and run in an offline fashion to generate data (including inputs to the model and the outputs (result of ED) that are dispatching signals for generation units). The data will be utilized in a PI-CNN to learn the feature. The learned PI-CNN will then be utilized for real-time decision-making. The model structure and training process of DNN and PI-CNN will be introduced in the following sections:

3.1. Model structure

The proposed structure of the convolutional neural network is shown in Fig. 2, which is elaborated in the following.

3.1.1. Convolutional layer

1D convolutional layer is used in our approach to extract the feature information from the 1D input data. Conv1D also reduces the huge training parameters's size as well as the training time. The operation of 1D convolutional layer is mathematically represented as:

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} w_{ik}^{l-1} \star o_i^{l-1} \tag{9}$$

x_k^l is the input data from economic dispatch (i.e., available powers of renewables, load power, and limits of generation units), b_k^l is the kth neuron bias at the lth layer, o_i^{l-1} is the ith neuron output at $l-1$th layer, w_{ik}^{l-1} is the kernel between the ith neuron at $l-1$th layer and kth neuron at lth layer, and $\star$ is the convolution operation. Fig. 3 shows the working process.

3.1.2. Activation function

In our approach, we choose the ReLU as the activation function. The equation representing the ReLU function is expressed by $f(x) = max(0, x)$, which means it returns 0 when the input is negative and keeps the original value when the input is positive. Because of this property, ReLU is considered as the most popular activation function. Comparing with the linear activation function, ReLU does not have the vanishing gradient problem and it is more efficient [35].

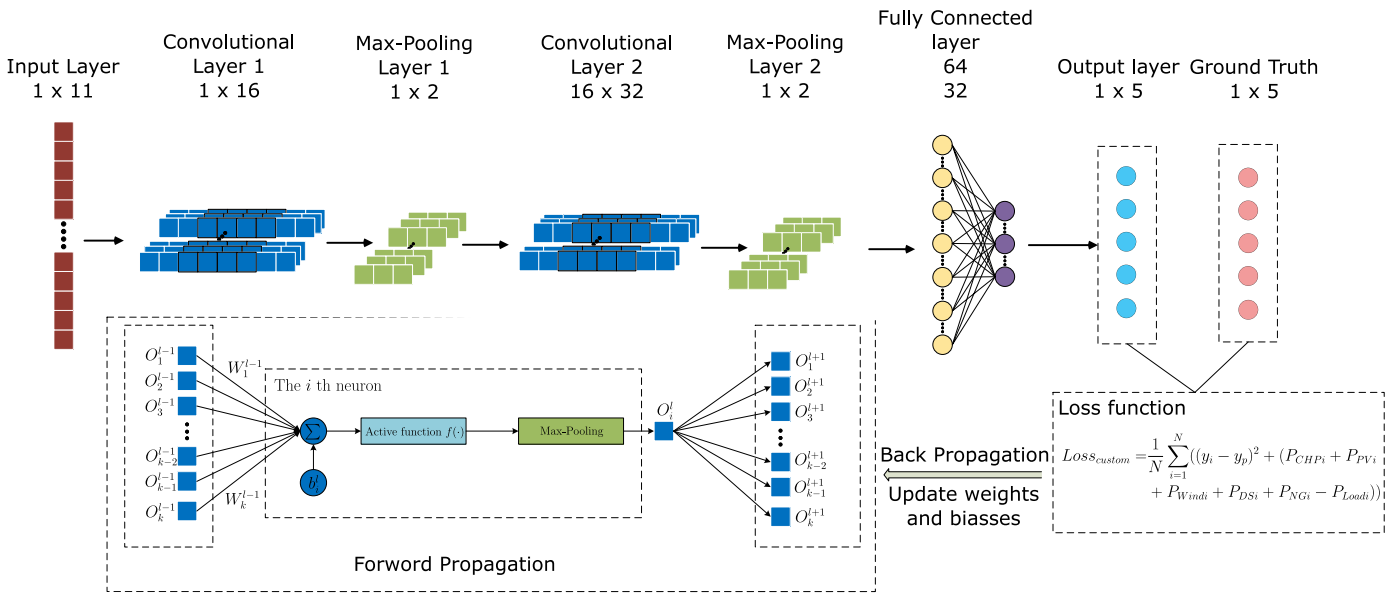


Fig. 2. Proposed PI-CNN model structure.

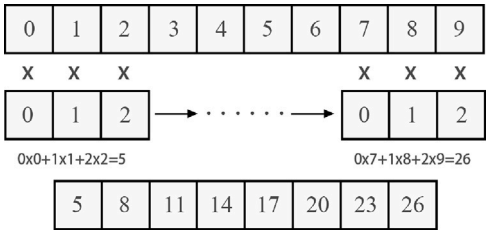


Fig. 3. Working process of 1-D Convolution layer.

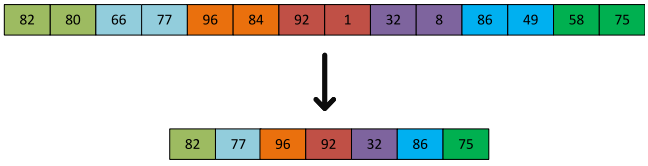


Fig. 4. Working process of max-pooling layer.

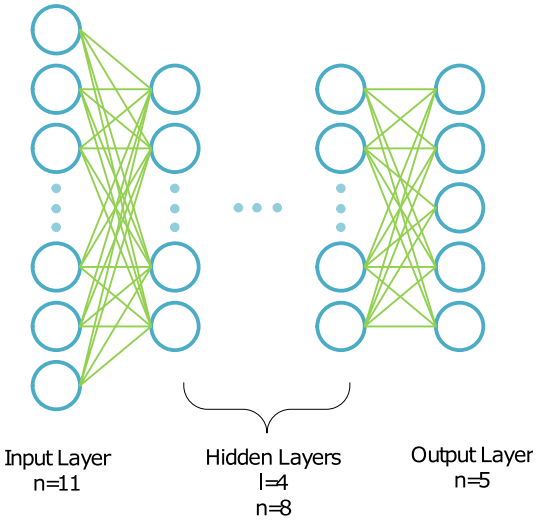


Fig. 5. DNN model structure.

3.1.3. MAX-Pooling layer

The max-pooling layers are used to avoid overfitting problems and improve the model's robustness by reducing the data dimensions and removing redundant information [36]. The working process is shown in Fig. 4

3.1.4. Fully connected layer

Fully connected layers with different locations (i.e., fully connected layers in the input or output layers) have different functions. The fully connected input layer takes the output of the previous layers, flattens them and turns them into a single vector that can be an input for the next process. The fully connected layers in the middle location take the inputs from the feature analysis and apply weights to predict the value. Fully connected output layer gives the final predicted values for each output.

The structure of DNN is shown in Fig. 5. There are 6 layers in the proposed DNN model, including 1 input layer, 1 output layer, and 4 hidden layers. Each hidden layer has 8 neurons. The optimizer and loss function are similar to those used in the CNN.

3.2. Loss function design

The loss function is the function to determine the distance between the model outputs and the ground truth. In the training process, the loss will reduce with the weights and bias updated in the back propagation process. Hence, to achieve the best performance, it is necessary to design targeted loss functions based on different tasks. Mostly, cross-entropy loss function is used for classification tasks and mean square error (MSE) loss function is utilized for regression tasks. Here, the MSE loss is chosen to be used for the ED problem. For the physics-inspired (PI) part, a constraint is added to the loss function, which acts as the prior knowledge of the loss function. It can help the model learn how to fit the observed data while forcing the prediction to fulfilled the given physical constraints. This kind of constraint can help one integrate more physical laws based on the constraints of the model into the formulation.

Table 1
Numerical evaluation criterion for CNN, DNN, SVM and TCN.

Numerical evaluation criterion															
	MSE					R^2					MAE				
Gen.	CNN	DNN-150	DNN-500	SVR	TCN	CNN	DNN-150	DNN-500	SVR	TCN	CNN	DNN-150	DNN-500	SVR	TCN
CHP	0.1517	0.1609	0.1112	0.1440	1.60517	0.9588	0.9563	0.9698	0.863133	0.626322	0.2185	0.2393	0.2327	0.274219	0.410143
NG	0.0333	0.2836	0.0390	0.1439	1.59023	0.9908	0.9216	0.9892	0.863166	0.623413	0.0409	0.3914	0.1146	0.271685	0.448424
DS	0.0133	0.4404	0.0503	0.1439	1.48704	0.9961	0.8719	0.9854	0.863238	0.62965	0.1397	0.4693	0.1616	0.264694	0.381698
Wind	1.3402	3.2092	3.1659	3.1860	19.6034	0.9942	0.9862	0.9885	0.985131	0.965832	0.6182	1.1664	1.0857	2.20896	2.41204
Sol	1.4872	2.4672	2.4479	2.4521	36.017	0.9973	0.9955	0.9968	0.968328	0.842155	0.5643	0.8716	0.8230	1.61966	3.15783

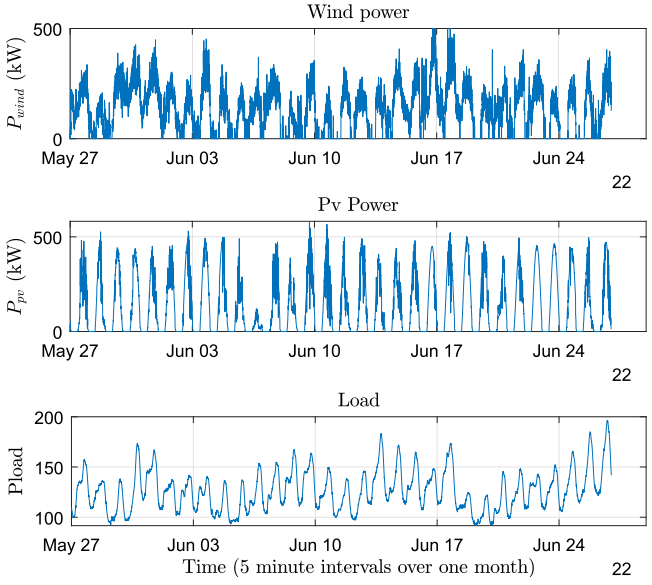


Fig. 6. Input data of P_{wind}, P_{Pv}, $Load$.

Table 2
Performance metrics of PI-CNN and CNN.

Numerical evalutaion criterion								
	MSE		R^2		MAE		$MAPE$	
Gen.	PI-CNN	CNN	PI-CNN	CNN	PI-CNN	CNN	PI-CNN	CNN
CHP	0.0379	0.1517	0.9897	0.9855	0.1581	0.2185	0.5484	0.45231
NG	0.0092	0.0333	0.9975	0.9908	0.0635	0.0049	0.3156	0.517303
DS	0.0093	0.0133	0.9973	0.9961	0.0851	0.1379	0.1136	0.385596
Wind	1.4370	1.3402	0.9938	0.9942	0.4978	0.6182	0.0542	0.0849615
Sol	1.4122	1.4872	0.9974	0.9973	0.4588	0.5643	0.2232	0.246068

3.2.1. MSE loss

The MSE loss is formulated as:

$$\mathcal{L}oss_{MSE} = \frac{1}{N}\|y_i - y_p\|_2^2 \tag{10}$$

where N is the batch size, y_i is the ground truth (solution of economic dispatch in kW) and y_p is the predicted value of the economic dispatch in kW.

3.2.2. Physics-informed loss

The goal of PI-loss is to aid the model learning the robust relationship between load and different generators described in (1b) to improve the model performance. To integrate the physical constraints into the CNN architecture, we first converted the inequality constraints into equality constraints by introducing slack variables. The equality constraints were then added to the loss function of the CNN. Additionally, to weight the impact of each constraint, weighting parameters λ_i are applied to the modified equality constraints, allowing us to adjust the influence of different constraints on the final loss function. We design the physics-informed loss function based on the MSE loss function and the power balance constraint (PBC), $\sum_{i=1}^{N_{G,i}} P_i^G(t) = P_{load}(t) \quad G \in \{CG, PV, W\}$. We add the power balance constrain $P_i^{CHP} + P_i^{PV} + P_i^W + P_i^{NG} - P_{Load} = \sum_{i=1}^{N_{G,i}} P_i^G - P_{load} = 0$ to the original MSE Loss function. The physics-informed loss function is defined as:

$$\begin{aligned}\mathcal{L}oss_{PI} &= \mathcal{L}oss_{MSE} + \mathcal{L}oss_{PBC} \\ &= \frac{1}{N}\|y_i - y_p\|_2^2 + \lambda_1 \| \sum_{i=1}^{N_{G,i}} P_{G,i} - P_{load}\|_2^2\end{aligned} \tag{11}$$

where λ_1 is a hyperparameter to penalize the solutions that violate the power balance. In addition, to ensure the solution of the proposed PI-based algorithm (i.e., generation dispatch) is always within the bounds and the ramp rate conditions are not violated, the constraints (1c)–(1g) will also need to be added to the loss function. To add the constraints to the loss function, we need to first convert the inequality constraints to equality constraints. Utilizing concepts from convex optimization theories, we first separate each constraint in (1c)–(1g) into two constraints and add a slack variable to each inequality constraint to convert it to an equality constraint suitable to be embedded in the loss function of the machine learning. The resulting constraints are written as

$$s_1 - P_i^G(t) = -U_{G,i}(t)P_{G,i}^{\min}, \tag{12}$$

$$s_2 + P_i^G(t) = U_{G,i}(t)P_{G,i}^{\max}, \tag{13}$$

$$s_3 + P_i^G(t) = P_{G,i}(t-1) + R_{\text{up}}^{\max}U_{G,i}(t-1) \tag{14}$$

$$s_4 - P_i^G(t) = R_{\text{down}}^{\max}U_{G,i}^t - P_i^G(t-1), \tag{15}$$

$$s_5 - P_i^{PV}(t) = -P_{PV,i}^{min} \tag{16}$$

$$s_6 + P_i^{PV}(t) = P_{PV,i}^{max} \tag{17}$$

$$s_7 - P_i^W(t) = -P_{W,i}^{min} \tag{18}$$

$$s_8 + P_i^W(t) = P_{W,i}^{max} \tag{19}$$

where s_1–s_6 are six slack variables defined to convert the inequality constraints to equality constraints. The new equality constraints in (12)–(19) can now be added to the loss function of the machine learning as

$$\mathcal{L}oss_{PI} = \mathcal{L}oss_{MSE} + \mathcal{L}oss_{PBC} + \mathcal{L}oss_{Con} \tag{20}$$

$$\begin{aligned}\mathcal{L}oss_{Con} = {} & \lambda_2\|s_1 - P_i^G(t) + U_{G,i}(t)P_{G,i}^{min}\|_2^2 + \\ & \lambda_3\|s_2 + P_i^G(t) - U_{G,i}(t)P_{G,i}^{max}\|_2^2 + \\ & \lambda_4\|s_3 + P_i^G(t) - P_{G,i}(t-1) - R_{\text{up}}^{\max}U_{G,i}(t-1)\|_2^2 + \\ & \lambda_5\|s_4 - P_i^G(t) - R_{\text{down}}^{\max}U_{G,i}^t + P_i^G(t-1)\|_2^2 + \\ & \lambda_6\|s_5 - P_i^{PV}(t) + P_{PV,i}^{min}\|_2^2 + \\ & \lambda_7\|s_6 + P_i^{PV}(t) - P_{PV,i}^{max}\|_2^2 + \\ & \lambda_8\|s_7 - P_i^W(t) + P_{W,i}^{min}\|_2^2 + \\ & \lambda_9\|s_8 + P_i^W(t) - P_{W,i}^{max}\|_2^2\end{aligned} \tag{21}$$

The combined loss function presented in (20) will ensure the constraints of the original economic dispatch function are embedded within the training of machine learning. It is also noted that $\lambda_2 - \lambda_9$ are new hyperparameters designed to penalize the violation of economic dispatch constraints and are tuned manually during the training process. Additionally, since the physics-informed loss function can accommodate hard and soft constraints into the lss function during the

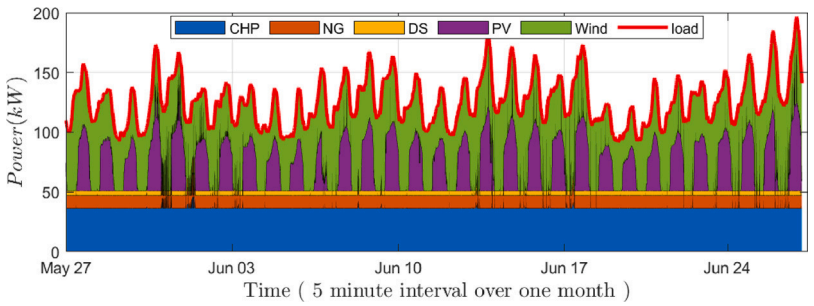


Fig. 7. Training datasets with output power from generation units and load.

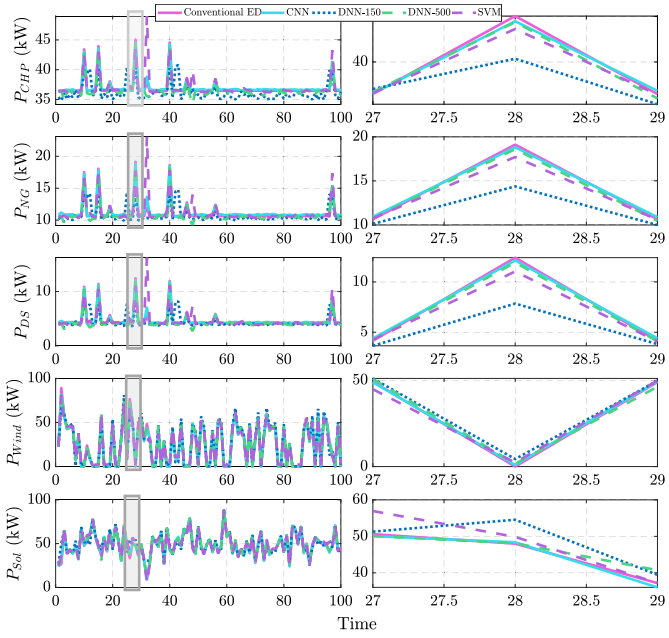


Fig. 8. Comparing the ED results of DNN and CNN.

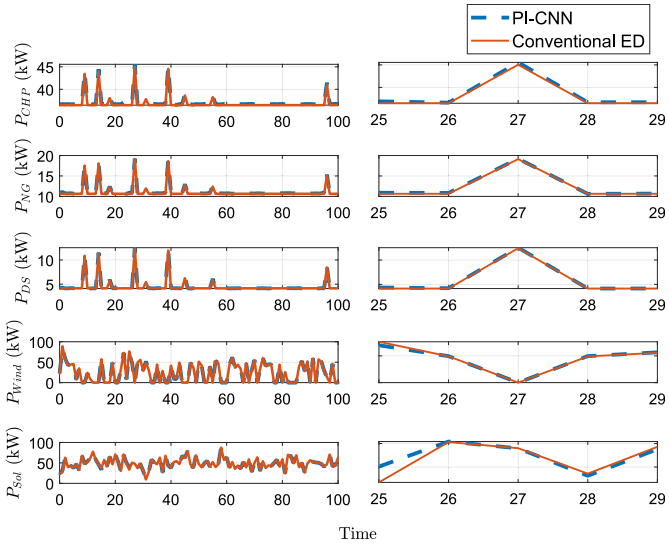


Fig. 9. The ED results of physics-inspired CNN.

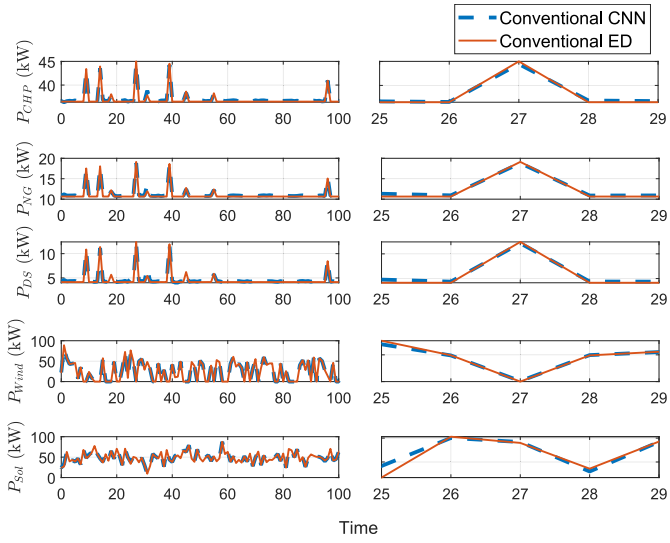


Fig. 10. The ED results of conventional CNN.

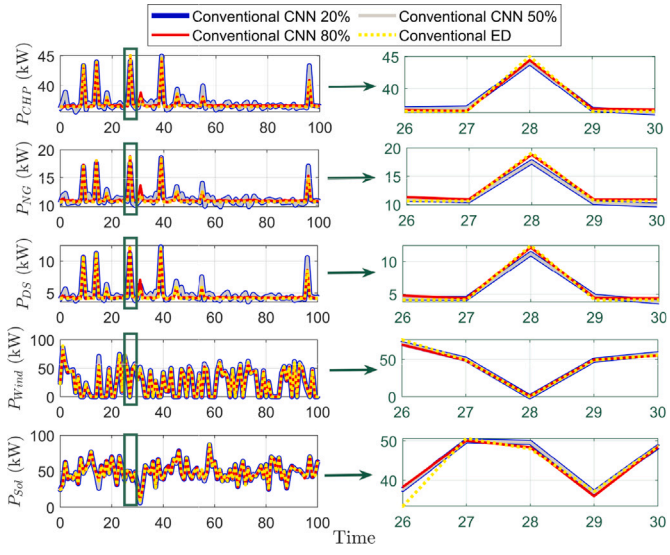


Fig. 11. Conventional CNN performance for different size of training data: 20%, 50% and 80%.

training of the proposed CNN model, the proposed PI-CNN model is not sensitive to the initial conditions like other PI-NN models [37]; [38].

3.3. PI-CNN working process

Forward propagation and back propagation are two parts of CNN's working process. The former involves sequentially calculating and preserving intermediate variables from the input layer to the output layer,

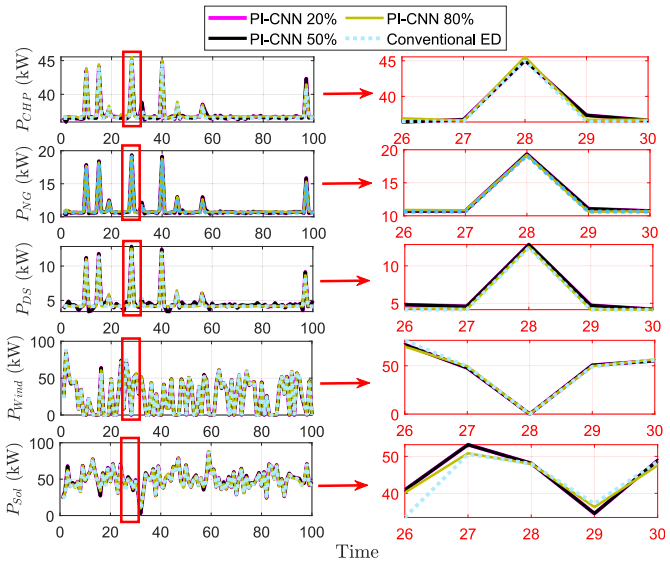


Fig. 12. PI-CNN performance for different size of training data: 20%, 50% and 80%.

as discussed above. Back propagation is a technique used for computing the gradient of CNN for updating the CNN parameters. Based on the chain rule from calculus, this method traverses the network in reverse order, starting from the output layer and moving towards the input layer. Its main purpose is to minimize the difference between predicted and actual output in order to update the parameters of the network [39]. The process from the output layer to the last convolutional layer can be mathematically presented as:

$$\frac{\partial \mathcal{L}}{\partial o_k^l} = \sum_{i=1}^{N_{l+1}} \frac{\partial \mathcal{L}}{\partial x_i^{l+1}} \frac{\partial x_i^{l+1}}{\partial o_k^l} = \sum_{i=1}^{N_{l+1}} \Delta_i^{l+1} w_{ki}^l \tag{22}$$

where $\mathcal{L}$ is the loss value. When the process proceeds to the next convolution layer, the delta error is:

$$\Delta_k^l = \frac{\partial \mathcal{L}}{\partial y_k^l} \frac{\partial y_k^l}{\partial x_k^l} = \frac{\partial \mathcal{L}}{\partial uo_k^l} \frac{\partial uo_k^l}{\partial y_k^l} f'(x_k^l) \tag{23}$$

where uo_k^l is the zero-order up sampling and $f(\cdot)$ is the activation function. The delta error through back propagation can be expressed as:

$$\Delta o_k^l = \sum_{i=1}^{N_{l+1}} \Delta_i^{l+1} \star R(w_{ki}^l) \tag{24}$$

where $R(\cdot)$ is the reverse function. Based on this, the sensitivities of weight and bias can be calculated as:

$$\frac{\partial \mathcal{L}}{\partial w_{ik}^l} = o_k^l \star \Delta_i^{l+1} \tag{25}$$

$$\frac{\partial \mathcal{L}}{\partial b_k^l} = \sum_n \Delta_k^l(n) \tag{26}$$

Then, the weight and bias can be updated by:

$$w_{ik}^{l-1}(t+1) = w_{ik}^{l-1}(t) - \varepsilon \frac{\partial \mathcal{L}}{\partial w_{ik}^{l-1}} \tag{27}$$

$$b_k^l(t+1) = b_k^l(t) - \varepsilon \frac{\partial \mathcal{L}}{\partial b_k^l} \tag{28}$$

4. Case studies

To verify the performance of the proposed CNN design, several case studies were conducted that include: (i) comparison among the performance of a regular CNN, conventional DNN, support vector machines (SVMs), and temporal convolutional neural networks (TCNs),

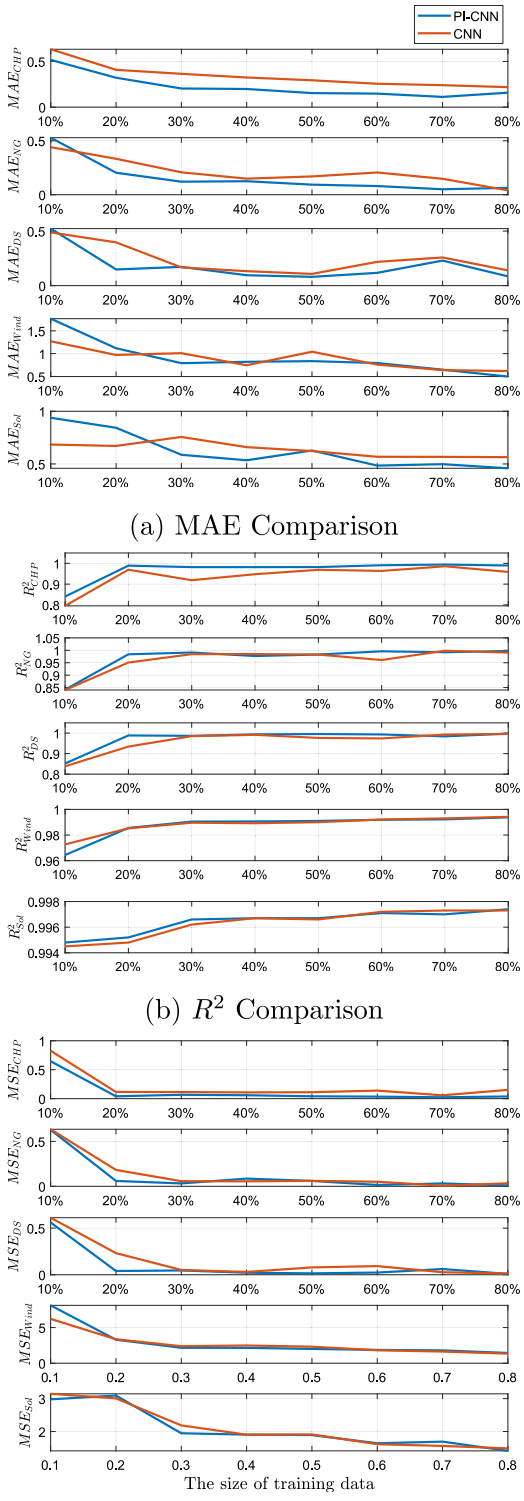


Fig. 13. Impact of size of training data on MSE, R^2, MAE values for CNN and PI-CNN.

(ii) comparison between the performance of a regular CNN and physics-informed CNN, (iii) the impact of the size of training data on physics-inspired CNN, (iv) runtime comparison, and (v) performance/feature comparison with the state-of-the-art machine learning-based microgrid ED. The case studies are conducted using torch library with PyTorch in

Python on a workstation with Intel Core i7-10870H CPU@2.20 GHZ, 16G RAM, and NVIDIA GeForce RTX-3070 GPU.

4.1. Datasets

The numerical solution of the economic dispatch (1a)–(1g) is generated by the OPTI Toolbox in MATLAB with "BONMIN" solver [40]. The dataset contains 8928 samples, each with 11 features (inputs) and 5 targets (generation unit dispatch), which represents the solution and constraints for one month of data at a 5-min resolution interval. The input data for wind power, solar PV power and load data for one month is shown in Fig. 6. Additional features in the dataset include the minimum and maximum limits of solar power and wind power, as well as the upper and lower bounds for CHP, natural gas power plant, and diesel power plant. The microgrid parameters and energy source limitations were sourced from [34]. The targets in this dataset refer to the power required from each energy source to fulfill the demand, which includes the power generated from solar, wind, CHP, natural gas, and diesel. The upper bound of the renewable generations is set as the maximum value of the prediction shown in Fig. 6. The complete training dataset, including output power from various generation units and load profiles, is displayed in Fig. 7.

4.2. Evaluation metrics

To accurately and comprehensively evaluate the proposed algorithm, four different evaluation metrics are implemented here: Coefficient of determination (R^2), mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error ($MAPE$).

4.2.1. R^2

The coefficient of determination (R^2) is a statistical metric that quantifies the proportion of variance in the dependent variables that can be explained by the independent variables in a regression model. Due to its dimensionless nature, R^2 facilitates the comparison of model performance across different datasets and various units of measurement. The mathematical representation of R^2 is shown below:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - y_{pi})^2}{\sum_{i=1}^{N}(y_i - y_m)^2} \tag{29}$$

where y_{pi} is the predicted value, y_i is the ground truth, and y_m is the mean value of the ground truth. The range of R^2 is from 0 to 1. The closer R^2 is to 1, the higher the model's fitness performance.

4.2.2. MSE

The mean squared error (MSE) measures the average squared difference between predicted and actual values. Its sensitivity to large errors makes it a robust measure of prediction accuracy, emphasizing discrepancies that are particularly undesirable. Moreover, it is widely used in prediction tasks, offering a consistent way to represent prediction accuracy. The function is shown below:

$$MSE = \frac{1}{N}\|y_i - y_{pi}\|_2^2 \tag{30}$$

4.2.3. MAE

Mean absolute error (MAE) represents the average absolute difference between the model predictive value and actual value. Different from MSE, MAE treats each error equally, providing a more balanced view of the model prediction performance.

$$MAE = \frac{1}{N}\|y_i - y_{pi}\| \tag{31}$$

Table 3
Comparison of training time and runtime of models.

Model	PI-CNN	CNN	DNN	Numerical
Training times (s)	640.297	349.495	345.402	NA
Running times (ms)	0.0414	0.0426	0.0330	20

4.2.4. MAPE

Mean absolute percentage error ($MAPE$) is a percentage-based metric that is independent of data scale, making it particularly useful for comparing prediction performance across datasets with varying magnitudes. The formula for calculating $MAPE$ is as follows:

$$MAPE = \frac{1}{N}\frac{|y_i - y_{pi}|}{y_i} \tag{32}$$

4.3. Data-prepossessing

Due to the considerable variation in the training data range, data normalization is applied before training. In this context, Min–max normalization is employed, which is formulated as:

$$X_\sigma = \frac{X_i - X_{min}}{X_{max-X_{min}}} \tag{33}$$

where X_{min}, X_{max} are the minimum and maximum of training data.

4.4. Results

This subsection aims to analyze the effectiveness of the proposed design in different case studies. The baseline setting for all case studies use 80% of data as the training data and 20% of the data as the testing data. All above performance metrics are used to provide a comprehensive performance evaluation.

4.4.1. Comparison between the CNN and other different machine learning methods

The aim of this case study is to compare the performance of the proposed CNN model and other typical machine learning methods, DNN, support vector machines (SVMs), and temporal convolutional neural networks (TCNs) to demonstrate the feasibility and efficiency of the proposed CNN model. Both CNN and DNN-150 were trained with 150 epochs, while DNN-500 had the same model structure as DNN-150 but was trained with 500 epochs. Because of the limitation of the SVM, it cannot deal with multi-output tasks, 5 different SVMs are designed for 5 different outputs. TCN is a typical deep neural network model that can deal with time-serious data, and here the history length is chosen as 100. The results obtained from different models are presented in Fig. 8, and the numerical evaluation results are illustrated in Table 1. From Fig. 8, it is observed that CNN has the best performance in fitting the ground truth compared with DNN-150, DNN-500, and SVM. Table 1 indicates that it is hard for DNN-150 and TCN to predict all the outputs from different generations accurately, while the proposed CNN model trained with the same number of epochs has a much better performance. SVMs perform better than DNN-150 and TCN. However, with the number of training epochs increasing, the performance of DNN will improve greatly, even better than SVMs. However, the MAE and MSE values of DNN-500 are still at least 6% less than the corresponding values for the CNN model, which showcases the feasibility and efficiency of the proposed CNN-based economic dispatch in accurately predicting the real-time dispatch of the generation units.

Table 4
Comparison between existing research on learning-based ED in microgrid.

Reference	Approaches	Deep learning	Physics inspired	Renewable generation	Best results
[20]	Random forest regression	No	No	Wind,PV	nRMSE: 6.35%
[21]	RBM	Yes	No	Wind	MAPE: 3.57%
[22]	FCRBM	Yes	No	None	MAPE: 0.4525%
[23]	DRNN	Yes	No	PV	MAE: 4.369
[24]	SPM	No	No	None	None numerical evaluation
[25]	Deep reinforcement learning	Yes	No	Wind,PV	None numerical evaluation
[26]	Deep reinforcement learning	Yes	No	Wind	None numerical evaluation
[7]	Reinforcement learning	No	No	Wind & PV	None
[27]	ANN+Q-Learning	Yes	No	2 renewable generations	nRMSE: 8.3%
[28]	Deep neural network	Yes	No	Wind	Exponential decay rate: 0.96
Proposed approach	PI-CNN	Yes	Yes	Wind, PV	MAE: 0.0635

4.4.2. Comparison between conventional CNN and physics-inspired CNN

Although the conventional CNN showed a high performance compared with the DNN in the last case study, it still has some limitations. More importantly, the prediction accuracy will decrease with the reduction of the training data size. The physics law embedded in the PI-CNN can make up those limitations. This case study aims to compare the performance of the CNN and PI-CNN, and show the advantages of PI-CNN.

Figs. 9 and 10 represent the prediction results, where the left column is the original results and the right column is the zoomed in version of the results. The numerical evaluation criterion is listed in Table 2. From the numerical evaluation of performance metrics, it is observed that under the base training condition (80% training data and 150 epochs), the PI-CNN has a much lower MSE for CHP, NG, DS, and solar and has a better performance than conventional CNN. In addition, Figs. 11 and 12 show the prediction results of conventional CNN and PI-CNN with different size of training data. The prediction accuracy reduces with reduction of the training data for both of those two models, while the PI-CNN shows the higher stability and efficiency when the size of the training data is smaller. Fig. 13 shows the MSE, R^2 and MAE values with different sizes of training data. From those three figures, both conventional CNN and PI-CNN have a stable high-quality performance when the size of training data is decreasing, even with only 20% training data, which proves the stability of the proposed CNN model. Furthermore, in most cases, PI-CNN has a better performance than the conventional CNN as is evident from the R^2 values, especially for the NG, with its R^2 value surpassing conventional CNN by up to 0.67%, and at least 6% lower in MSE and MAE value under the same training condition.

4.5. Training time and running time for all models

This case study aims to provide a comprehensive comparison of the computational efficiency exhibited by the studied models. Therefore, we thoroughly investigated the training and running times of each model under consistent training conditions. Specifically, 80% of the data was used for the training and 20% of the data was used for testing in all models. In addition, all the algorithms were run for 150 training epochs. The results are presented in Table 3.

It is observed that when compared to the conventional CNN, the training time of the PI-CNN model is found to be 1.8 times longer under identical training conditions. This divergence in training times can be attributed to the increased computational complexity caused by the physical laws embedded in the loss function of PI-CNN, which can be regarded as a drawback of the PI-CNN model. However, in pursuit of superior prediction performance, sacrificing a small portion of training time can be deemed a better choice. Conversely, the training time of the DNN model remains nearly the same as the conventional CNN. In addition, although the training time of the PI-CNN is larger compared with the other two approaches, the runtimes of PI-CNN and CNN are fairly close (0.0414 and 0.0426 ms, respectively) on the testing dataset, indicating no computational complexity for close to real-time execution of ED problem with PI-CNN. The DNN has a shorter runtime (0.0330 ms), but given the lower accuracy of DNN compared with PI-CNN or CNN, one finds a trade-off between the runtime of the machine learning-based ED and the accuracy of the prediction. Besides those, the prediction runtime for the numerical optimization method (conventional ED) is 20 ms. The runtime of numerical optimization is around 480 times slower than deep learning models, and the difference increases with the complexity size of the system.

4.6. Comparison with the state-of-the-art machine learning-based ED

This case study compares the existing research on machine learning-based microgrid ED with the proposed approach. Table 4 illustrates the features of the existing approaches [20–28] for solving the ED problem in microgrid.

As it can be seen, existing approaches to address the uncertainties stemming from intermittent renewable generation in ED problems primarily rely on deep learning-based methods, while our proposed CNN approach is more accurate and requires less training epochs. In addition, none of these approaches utilize physics-inspired machine learning for solving the ED problem. In contrast, our proposed approach combines the physical laws with deep learning, presenting a novel contribution to the field.

5. Conclusion

This paper presents a novel approach using physics-informed convolutional neural network for solving the economic dispatch problem in Microgrids. By incorporating the constraints of a numerical economic dispatch problem into the learning process, the machine learning algorithm can be better trained to predict the solution of ED problem in real-time without solving the conventional computationally complex numerical optimization. The performance of the proposed approach is compared with DNN-based and conventional CNN-based approaches using several deterministic case studies. Results demonstrate that the PI-CNN approach achieves excellent results with a small training dataset (even with 80% less data) and runs very fast (0.04 ms), which is extremely faster than conventional numerical optimization-based solutions (400 times faster). A limitation of the proposed PI-CNN is its reliance on the quality of input data to produce accurate outputs, i.e., available output power of renewable generation highly depends on the weather data. Future work will focus on increasing the model's stability with respect to weather data, as well as enhancing the design with a closed-loop economic dispatch engine that integrates dynamic models of assets to improve responsiveness to microgrid events.

Funding

This work was in part under support from the Department of Defense, Office of Naval Research award number N00014-23-1-2602 and was financed (in part) by a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development, through the Pennsylvania Infrastructure Technology Alliance (PITA) grant.

CRediT authorship contribution statement

Xiaoyu Ge: Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Data curation. **Javad Khazaei:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Javad Khazaei reports financial support and article publishing charges were provided by US Department of Defense. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

[1] L. Wang, X. An, H. Xu, Y. Zhang, Multi-agent-based collaborative regulation optimization for microgrid economic dispatch under a time-based price mechanism, Electr. Power Syst. Res. 213 (2022) 108760.
[2] Y. Duan, Y. Zhao, J. Hu, An initialization-free distributed algorithm for dynamic economic dispatch problems in microgrid: Modeling, optimization and analysis, Sustain. Energy Grids Netw. 34 (2023) 101004.
[3] P.M. Joshi, H. Verma, An improved TLBO based economic dispatch of power generation through distributed energy resources considering environmental constraints, Sustain. Energy Grids Netw. 18 (2019) 100207.
[4] W. Sun, Y. Qiao, W. Liu, Economic scheduling of mobile energy storage in distribution networks based on equivalent reconfiguration method, Sustain. Energy Grids Netw. 32 (2022) 100879.
[5] M. Akhlaghi, Z. Moravej, A. Bagheri, Flexible and sustainable scheduling of electric power grids: A dynamic line and transformer rating based approach under uncertainty condition, Sustain. Energy Grids Netw. 36 (2023) 101150.
[6] Y. Li, T. Li, H. Zhang, X. Xie, Q. Sun, Distributed resilient double-gradient-descent based energy management strategy for multi-energy system under DoS attacks, IEEE Trans. Netw. Sci. Eng. 9 (4) (2022) 2301–2316.
[7] L. Yang, X. Li, M. Sun, C. Sun, Hybrid policy-based reinforcement learning of adaptive energy management for the energy transmission-constrained island group, IEEE Trans. Ind. Inform. 19 (11) (2023) 10751–10762.
[8] Y. Li, H. Zhang, X. Liang, B. Huang, Event-triggered-based distributed cooperative energy management for multienergy systems, IEEE Trans. Ind. Inform. 15 (4) (2018) 2008–2022.
[9] Y. Li, J. Wang, D. Zhao, G. Li, C. Chen, A two-stage approach for combined heat and power economic emission dispatch: Combining multi-objective optimization with integrated decision making, Energy (Oxford) 162 (2018) http://dx.doi.org/10.1016/j.energy.2018.07.200, URL: https://www.osti.gov/biblio/1490180.
[10] H. Hou, M. Xue, Y. Xu, Z. Xiao, X. Deng, T. Xu, P. Liu, R. Cui, Multi-objective economic dispatch of a microgrid considering electric vehicle and transferable load, Appl. Energy 262 (2020) 114489.
[11] H. Narimani, S.-E. Razavi, A. Azizivahed, E. Naderi, M. Fathi, M.H. Ataei, M.R. Narimani, A multi-objective framework for multi-area economic emission dispatch, Energy 154 (2018) 126–142, http://dx.doi.org/10.1016/j.energy.2018.04.080, URL: https://www.sciencedirect.com/science/article/pii/S036054421830687X.
[12] Z. Zhang, D. Yue, C. Dou, H. Zhang, Multiagent system-based integrated design of security control and economic dispatch for interconnected microgrid systems, IEEE Trans. Syst. Man Cybern. Syst. 51 (4) (2020) 2101–2112.
[13] Y.-Y. Lee, R. Baldick, A frequency-constrained stochastic economic dispatch model, IEEE Trans. Power Syst. 28 (3) (2013) 2301–2312, http://dx.doi.org/10.1109/TPWRS.2012.2236108.
[14] Y. Xu, Z. Dong, Z. Li, Y. Liu, Z. Ding, Distributed optimization for integrated frequency regulation and economic dispatch in microgrids, IEEE Trans. Smart Grid 12 (6) (2021) 4595–4606.
[15] Z. Lin, H. Chen, Q. Wu, W. Li, M. Li, T. Ji, Mean-tracking model based stochastic economic dispatch for power systems with high penetration of wind power, Energy 193 (2020) 116826, http://dx.doi.org/10.1016/j.energy.2019.116826, URL: https://www.sciencedirect.com/science/article/pii/S0360544219325216.
[16] B. Huang, L. Liu, H. Zhang, Y. Li, Q. Sun, Distributed optimal economic dispatch for microgrids considering communication delays, IEEE Trans. Syst. Man Cybern. Syst. 49 (8) (2019) 1634–1642.
[17] H. Shuai, J. Fang, X. Ai, Y. Tang, J. Wen, H. He, Stochastic optimization of economic dispatch for microgrid based on approximate dynamic programming, IEEE Trans. Smart Grid 10 (3) (2018) 2440–2452.
[18] W.-C. Yeh, M.-F. He, C.-L. Huang, S.-Y. Tan, X. Zhang, Y. Huang, L. Li, New genetic algorithm for economic dispatch of stand-alone three-modular microgrid in dongao island, Appl. Energy 263 (2020) 114508.
[19] H. Hou, M. Xue, Y. Xu, Z. Xiao, X. Deng, T. Xu, P. Liu, R. Cui, Multi-objective economic dispatch of a microgrid considering electric vehicle and transferable load, Appl. Energy 262 (2020) 114489.
[20] W. Dong, Q. Yang, W. Li, A.Y. Zomaya, Machine-learning-based real-time economic dispatch in islanding microgrids in a cloud-edge computing environment, IEEE Internet Things J. 8 (17) (2021) 13703–13711.
[21] A. Kalakova, H.K. Nunna, P.K. Jamwal, S. Doolla, A novel genetic algorithm based dynamic economic dispatch with short-term load forecasting, IEEE Trans. Ind. Appl. 57 (3) (2021) 2972–2982.
[22] G. Hafeez, K.S. Alimgeer, I. Khan, Electric load forecasting based on deep learning and optimized by heuristic algorithm in smart grid, Appl. Energy 269 (2020) 114915.
[23] L. Wen, K. Zhou, S. Yang, X. Lu, Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting, Energy 171 (2019) 1053–1065.
[24] W. Gil-González, O.D. Montoya, E. Holguín, A. Garces, L.F. Grisales-Noreña, Economic dispatch of energy storage systems in dc microgrids employing a semidefinite programming model, J. Energy Storage 21 (2019) 1–8.
[25] Z. Liu, Y. Liu, H. Xu, S. Liao, K. Zhu, X. Jiang, Dynamic economic dispatch of power system based on DDPG algorithm, Energy Rep. 8 (2022) 1122–1129.
[26] S. Zhou, Z. Hu, W. Gu, M. Jiang, M. Chen, Q. Hong, C. Booth, Combined heat and power system intelligent economic dispatch: A deep reinforcement learning approach, Int. J. Electr. Power Energy Syst. 120 (2020) 106016.
[27] R. Rocchetta, L. Bellani, M. Compare, E. Zio, E. Patelli, A reinforcement learning framework for optimal operation and maintenance of power grids, Appl. Energy 241 (2019) 291–301, http://dx.doi.org/10.1016/j.apenergy.2019.03.027, URL: https://www.sciencedirect.com/science/article/pii/S0306261919304222.
[28] Y. Du, F. Li, Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning, IEEE Trans. Smart Grid 11 (2) (2019) 1066–1076.
[29] 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.
[30] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422–440.
[31] S. Harbola, V. Coors, One dimensional convolutional neural network architectures for wind prediction, Energy Convers. Manage. 195 (2019) 70–75.
[32] F. Moazeni, J. Khazaei, J.P.P. Mendes, Maximizing energy efficiency of islanded micro water-energy nexus using co-optimization of water demand and energy consumption, Appl. Energy 266 (2020) 114863.
[33] M. Hijjo, F. Felgner, G. Frey, PV-battery-diesel microgrid layout design based on stochastic optimization, in: 2017 6th International Conference on Clean Electrical Power, ICCEP, IEEE, 2017, pp. 30–35.
[34] F. Moazeni, J. Khazaei, A. Asrari, Step towards energy-water smart microgrids; buildings thermal energy and water demand management embedded in economic dispatch, IEEE Trans. Smart Grid 12 (5) (2021) 3680–3691.
[35] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.
[36] M. Afrasiabi, M. Mohammadi, M. Rastegar, A. Kargarian, Probabilistic deep neural network price forecasting based on residential load and wind speed predictions, IET Renew. Power Gener. 13 (11) (2019) 1840–1848.
[37] Y. Nakamura, S. Shiratori, H. Nagano, K. Shimano, Physics-informed neural network with variable initial conditions, in: Proceedings of the 7th World Congress on Mechanical, Chemical, and Material Engineering, Prague, Czech Republic, 2021, pp. 2–4.
[38] Y. Zong, Q. He, A.M. Tartakovsky, Improved training of physics-informed neural networks for parabolic differential equations with sharply perturbed initial conditions, Comput. Methods Appl. Mech. Engrg. 414 (2023) 116125.
[39] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D.J. Inman, 1D convolutional neural networks and applications: A survey, Mech. Syst. Signal Process. 151 (2021) 107398.
[40] J. Currie, D.I. Wilson, N. Sahinidis, J. Pinto, OPTI: Lowering the barrier between open source optimizers and the industrial MATLAB user, Found. Comput.-Aided Process Oper. 24 (2012) 32.