energies

Review

Deep Neural Networks in Power Systems: A Review

Mahdi Khodayar *,† and **Jacob Regan** †

Department of Computer Science, University of Tulsa, Tulsa, OK 74104, USA; jwr7678@utulsa.edu
* Correspondence: mahdi-khodayar@utulsa.edu
† These authors contributed equally to this work.

Abstract: Identifying statistical trends for a wide range of practical power system applications, including sustainable energy forecasting, demand response, energy decomposition, and state estimation, is regarded as a significant task given the rapid expansion of power system measurements in terms of scale and complexity. In the last decade, deep learning has arisen as a new kind of artificial intelligence technique that expresses power grid datasets via an extensive hypothesis space, resulting in an outstanding performance in comparison with the majority of recent algorithms. This paper investigates the theoretical benefits of deep data representation in the study of power networks. We examine deep learning techniques described and deployed in a variety of supervised, unsupervised, and reinforcement learning scenarios. We explore different scenarios in which discriminative deep frameworks, such as Stacked Autoencoder networks and Convolution Networks, and generative deep architectures, including Deep Belief Networks and Variational Autoencoders, solve problems. This study's empirical and theoretical evaluation of deep learning encourages long-term studies on improving this modern category of methods to accomplish substantial advancements in the future of electrical systems.

Keywords: deep learning; power systems; discriminative neural networks; generative modeling

check for updates

Citation: Khodayar, M.; Regan, J. Deep Neural Networks in Power Systems: A Review. *Energies* **2023**, *16*, 4773. https://doi.org/10.3390/en16124773

Academic Editors: Mateus Mendes and Inácio Fonseca

Received: 18 May 2023
Revised: 5 June 2023
Accepted: 6 June 2023
Published: 17 June 2023

1. Introduction

The precision and dependability of data-driven approaches employed for the management and assessment of power systems are highly dependent on the choice of the data representation (i.e., properties derived from the source data) [1]. Consequently, the majority of issues regarding the use of traditional data-driven algorithms for power systems are centered on the layout of approaches to preprocessing employing unsupervised dimensionality reduction methods, such as principal component analysis (PCA) [2], linear discriminant analysis (LDA) [3], and t-distributed stochastic neighbor embedding (t-SNE) [4]. Such methods for discovering features substantially raise the computational and memory complexities of algorithms relying on data and lead to inadequate precision as they are unable to detect extremely complex and highly fluctuating patterns within a data set's general domain.

Current artificial intelligence (AI) investigations on wind prediction [5–7], photovoltaic (PV) energy forecasting [8–10], state estimation [11,12], electrical grid generation [13,14], and energy segmentation [15,16] demonstrate that creating models fueled by data with less reliance on explicit ways of preprocessing (e.g., PCA) results in significantly improved prediction and regression quality. In this context, shallow artificial neural networks (ANNs) with few computational layers are proposed for load forecasting [17], probabilistic wind and solar power generation prediction [18], economic dispatch [19], voltage stability monitoring [20], system identification [21], nonlinear power system excitation control [22], and inductive coupling between overhead power lines and nearby metallic pipelines [23].

Shallow neural networks have several drawbacks compared to deep neural networks. One of the main limitations is their limited representational power. Shallow ANNs are merely capable of learning semi-linear decision boundaries, which restricts their ability to handle highly nonlinear and complex patterns present in many real-world problems [24].

Additionally, shallow networks lack the ability to learn hierarchical representations of features, making it challenging to capture abstract and high-level concepts. However, deep learning, enabled by deep neural networks with multiple hidden layers, has addressed these issues in recent studies [25]. Deep neural networks can learn highly nonlinear mappings and capture intricate relationships in the data. The multiple layers in deep networks enable the automatic extraction of hierarchical features, allowing them to capture both low-level and high-level abstractions [26]. In lieu of a formal preprocessing strategy, deep learning studies form a multi-layer ANN with more than one hidden layer composed of several complex computational activations [24]. The deep ANN variables (i.e., the weights and biases) are usually learned via a greedy unsupervised or semisupervised step-by-step manner [27], in which each layer extracts dynamic features from the features calculated from the preceding layer.

On the basis of conceptual considerations, deep learning methods suggested for use in smart grid-related purposes are typically classified into three main categories:

(1) Discriminative methods: The objective of discriminative ANNs is to instantly acquire a highly complex decision boundary between distinct classes as well as regression patches in the power grid measurement datasets [28]. The rectified linear unit (ReLU) ANN [29] proposed for immediate reliability management response is highlighted in this group of models. Because of its high adaptation potential and low computational burden, the ReLU ANN is used for real-time small signal stability evaluation [30], defective line location [31], and phasor measurement unit (PMU)-driven incident categorization [32]. In addition, the Stacked Autoencoder (SAE) has been proposed as an intensely nonlinear variant of the PCA for unsupervised identification of statistical patterns and structures in wind energy forecasting [33], solar energy estimation [34], fault classification [35], and transient stability assessment [36,37]. In addition, the Long Short-Term Memory (LSTM) neural architecture is proposed as a supervised time-related feature extraction technique with a deep recursive structure to model the ordered pattern of time-varying power system observations [38–40]. LSTM-based ordered models have been put forward for wind and solar generation predictions [41], demand modeling utilizing universal statistics [42], immediate power fluctuation recognition [43], electricity consumption prediction [44], energy decomposition into electrical devices [45], sustainable energy prediction [46], as well as fault recognition [47]. Because of the nature of their convolutional and pooling procedures, Convolutional Neural Network (CNN) models are highly effective at capturing patterns of coherence in power system measurements [48]. Applying predictive convolutional filters, CNN identifies significant spatial and temporal relationships among observations [49]. The combination of pooling and convolutional layers in this form of artificial neural networks encompasses the location of measurements into their chronological attributes to achieve spatiotemporal objectives in the field of sustainable energy prediction [50], transient stability evaluation [51], harmonic components assessment [52], fault identification [53], and short-term voltage stability inspection [54].

(2) Probabilistic deep artificial neural networks view feature modeling as a technique for discovering a minimal set of concealed factors that best characterizes the probability density function (PDF) of the information being studied [55]. The PDF is subsequently converted to the problem's destination discrete class or real number. Deep Belief Network (DBN) is a widely recognized stochastic graphical system within this group that discovers the PDF of the data set considering its conditionally independent hidden variables [56]. In this framework, Gibbs sampling is usually employed to acquire the features that are necessary to offer an accurate estimate of the stochastic patterns of the provided data for uncertain systems that must account for massive uncertainty elements. Wind and solar energy forecasting [57], transient stability evaluation [58], long-term and mid-term demand forecasting [59], and stochastic power grid state estimation [60] are the primary applications of DBN-based techniques. In addition, in this

class of frameworks, the Generative Adversarial Network (GAN) is presented, which takes samples from an estimated PDF and contrasts them with the actual information in the source data in order to improve the preciseness of the taught PDF. This approach has recently been applied to significant anomaly and fault identification problems for small-sample turbines [61] and distributed energy system cybersecurity [62] due to its ability to effectively acquire the primary features of the PDF. Moreover, because GANs can generate datasets using samples derived from the predicted PDF, these models have recently been applied to model-free sustainable energy story synthesis challenges [63]. Variational Autoencoder (VAE) is introduced as an innovative variant of deep generative artificial neural networks that discover the PDF of the collected data through the discovery of a high-dimensional hidden variable that is linked to the raw data examples in the original dataset. It has been demonstrated that VAE produces accurate artificial data for power grid simulation [64], unsupervised identification of anomalies in energy historical datasets [13], and Electric Vehicle demand production [65].

(3) Deep Reinforcement Learning (DRL) methods have been an important category of machine learning techniques that aim to discover the best policy for continuous/discrete action selection based on the environment's input (i.e., objective) calculated through a function that rewards intelligent agents' behavior. This function indicates, according to the present condition of the system, the degree to which the machine learning task's objective and constraints have been met. As opposed to traditional deep learning, which estimates a discrete objective function for classifiers and a real-valued objective function for regression, DRL seeks to reduce a generic objective function specified by the training scenario in a completely visible or partially visible setting. Consequently, this technique solves more comprehensive groups of problems than traditional deep learning. Due to its reward-based structure, DRL is extensively used for a variety of control challenges, such as voltage control [66], adaptable emergency control [67], and self-learning control for energy-efficient transportation [68]. DRL is also applied to optimization tasks for learning ideal pricing techniques in electricity markets [69], demand response plans for the management of energy [70–72], and determining the best wind and storage collaborative scheme to reduce the impact of unpredictability in sustainable energy production in electric networks [73]. Moreover, the detection and classification of cybersecurity threats [74], dynamic energy allocation [75], and power grid data security protection [73] are recent applications of this group of techniques.

In the area of power grid research, this review paper investigates three of the main types of deep neural networks. Section 2 introduces the deep discriminative architectures and their training procedure. Employing several practical applications and power systems datasets, this paper explains and compares analytically and empirically various variants of this machine learning category of approaches. In Section 3, probabilistic deep neural architectures and techniques including the traditional DBN and its Gaussian variant, in addition to the recently developed GANs and VAEs, are introduced. In this section, the practical and conceptual benefits of these methods are presented. The paper then examines DRL algorithms and their broad use in power network operation and management in Section 4. Section 5 presents the emerging topics and new challenges in the area of deep learning. Finally, Section 6 concludes with a discussion of the findings and future machine learning tasks in this domain.

2. Discriminative Deep Architectures

Discriminative machine learning is a powerful tool for solving supervised problems in which the goal is to learn a mapping between input features and output labels. In this approach, we aim to directly model the conditional probability distribution of the output given the input, also known as the posterior probability, using a discriminative function [76]. In mathematical terms, given a training set of input-output pairs $(x_1, y_1), \ldots, (x_n, y_n)$, the goal is to learn a function $f(x)$ that predicts the output label y given an input x.

The discriminative function $f(x)$ is typically learned by optimizing a loss function that measures the difference between the predicted output and the true output [77]. One common approach for discriminative learning is to use logistic regression, which models the posterior probability of the output label given the input features as a logistic function $P(y|x) = \sigma(w^T x + b)$ where $\sigma(z)$ is the sigmoid function, w is a vector of weights, b is a bias term, and x is a vector of input features [78]. The sigmoid function maps the output of the linear function $w^T x + b$ to the range $[0, 1]$, which can be interpreted as a probability. The logistic regression model can be trained using maximum likelihood estimation [79], which involves maximizing the log-likelihood of the training data with n training samples $L(w, b) = \sum_{i=1}^{n} logP(y_i|x_i; w, b)$ where $P(y_i|x_i; w, b)$ is the posterior probability of the output label y_i given the input x_i corresponding to sample i in the dataset, parameterized by the weights w and bias b.

The multilayer perceptron (MLP) is a type of discriminative ANN that consists of multiple layers of neurons, each layer being fully connected to the previous and next layers. The MLP is a powerful tool for solving a wide range of supervised learning problems, including classification, regression, and pattern recognition [80]. Mathematically, an MLP can be represented as a function $f(x)$ that maps an input vector x to an output vector y, where y is a function of the weighted sum of the activations of the neurons in the previous layer. Let us consider a feedforward MLP with L layers, where the input layer has d input neurons, the output layer has k output neurons, and each hidden layer has m neurons. The output of the j-th neuron in the l-th layer, denoted as $a_j^{(l)}$, is computed as $a_j^{(l)} = g\left(\sum_{i=1}^{m^{(l-1)}} w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right)$ where $g(z)$ is an activation function, $w_{ij}^{(l)}$ is the weight connecting the i-th neuron in layer $l-1$ to the j-th neuron in layer l, $b_j^{(l)}$ is the bias term for the j-th neuron in layer l, and $m^{(l-1)}$ is the number of neurons in the previous layer. The activation function $g(z)$ introduces nonlinearity to the network, allowing it to learn complex mappings between the input and output.

To train an MLP, we need to minimize a loss function that measures the difference between the predicted output and the true output. One common loss function is the mean squared error (MSE) [81], which is defined as $\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} |y_i - f(x_i)|^2$ where n is the number of training samples, y_i is the true output for the i-th sample, and $f(x_i)$ is the predicted output for the i-th sample. To minimize the loss function, one can use stochastic gradient descent (SGD) [26], which updates the weights and biases of the network in the direction of the negative gradient of the loss function with respect to the weights and biases. The weight update rule for the jth neuron in the lth layer is given by $w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}}$ where η is the learning rate, which controls the step size of the weight update, and $\frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}}$ is the partial derivative of the loss function with respect to the weight $w_{ij}^{(l)}$.

The vanishing gradient problem is a common issue in training deep multilayer perceptrons, as the gradients of the loss function with respect to the weights in the lower layers tend to become very small as they propagate backward through the network [46]. This can cause the weights in the lower layers to be updated very slowly, or not at all, leading to poor performance. Deep learning methods address this issue by introducing specialized activation functions, weight initialization schemes, and optimization techniques that are specifically designed to prevent the gradients from vanishing or exploding [24]. For example, activation functions like ReLU and variants of it have a non-zero derivative in most regions of their domain, which helps to mitigate the vanishing gradient problem. Additionally, weight initialization schemes such as He initialization [82] and optimization techniques such as adaptive learning rate methods (e.g., Adam) [83] have also been shown to improve training performance in deep neural networks. These techniques have enabled the training of very deep neural networks with many layers, allowing them to learn complex patterns and representations in the data.

2.1. ReLU Neural Networks

ReLU neural networks are a class of deep neural networks that use the ReLU activation function in their hidden layers. ReLU is a piecewise linear function that returns the input value if it is positive, and zero otherwise. It has become a popular choice for neural networks due to its simplicity, computational efficiency, and effectiveness in preventing the vanishing gradient problem. One of the main advantages of ReLU activation functions is that they have a non-zero derivative for all positive inputs, which helps to prevent the vanishing gradient problem. Additionally, the ReLU function is computationally efficient to evaluate, as it only involves a simple thresholding operation. This makes it well-suited for large-scale neural networks that require many activations to be computed during training and inference. Another advantage of ReLU neural networks is their ability to learn sparse representations of the input data [84]. Since the ReLU function sets negative values to zero, the activations of many neurons in the network will be zero for most input samples, resulting in a sparse representation. This can lead to faster training and improved generalization performance, as the network focuses on the most relevant features of the data. However, ReLU activation functions are not without their drawbacks. One issue is that they can suffer from the "dying ReLU" problem [85], where the gradient of the function is zero for all negative inputs, causing the neuron to stop learning. This can happen if the weights of the neuron are initialized such that it only receives negative inputs. To address this issue, various modifications to the ReLU function have been proposed, such as leaky ReLU [86], which adds a small slope to the negative part of the function.

Table 1 shows the applications of discriminative deep neural networks for power systems operation, management, and planning. Due to their high generalization power, deep ReLU networks are widely applied in power systems classification and regression problems. For instance, as shown in Table 1, Deep ReLU networks result in accurate voltage stability assessments with a root mean squared error (RMSE) of 0.083 and mean absolute percentage error (MAPE) equal to 0.095 on the New England 10-generator 39-bus system [29]. In addition, Deep ReLU models achieve an RMSE of 0.045 and MAPE of 0.083 for power system reliability assessment of the IEEE-RTS-79 system [87]. Furthermore, this type of deep learning model is employed for load identification and estimation of complex load parameters. In this area, the deep ReLU network is applied to the loads of the 68-bus New England and New York Interconnect System, which yields an RMSE of 0.045 and a MAPE of 0.074 in real-time load modeling tasks [88]. Since the feedforward algorithm of these neural networks takes merely several milliseconds, these approaches can be easily tested on real-world power systems in a real-time fashion. The use of this approach for sustainable energy prediction results in high accuracy and reliability for both wind and solar energy prediction tasks [89,90]. The RMSE and MAPE of the deep ReLU network for hourly wind power prediction are 0.078 and 0.092, respectively. These results are reported for the Wind Integration National Dataset. Additionally, the RMSE and MAPE of this approach for the hourly photovoltaic power prediction are 0.093 and 0.104 using the Solar Power Data for Integration Studies. Energy disaggregation is another application in which the deep ReLU network shows high performance. This technique results in an F-score of 68.72 with a precision equal to 80.54 on the Reference Energy Disaggregation Dataset [16]. Similar high classification accuracies are reported for fault identification [91]. For instance, the use of deep ReLU networks for fault detection in the New England 39-bus test system results in an F-score of 77.49 which is a reasonable accuracy for real-world scenarios.

Table 1. Discriminative deep neural networks in power system studies.

Application	Dataset	Model	Performance Metric	Result
Voltage Stability Assessment [28,29,92,93]	New England 10-generator 39-bus system	ReLU	RMSE	0.083
			MAPE	0.095
		SAE	RMSE	0.071
			MAPE	0.086
		LSTM	RMSE	0.035
			MAPE	0.046
		CNN-LSTM	RMSE	0.021
			MAPE	0.038
Power System Reliability Evaluation [38,87,94,95]	IEEE-RTS-79 system	ReLU	RMSE	0.045
			MAPE	0.083
		SAE	RMSE	0.032
			MAPE	0.071
		CNN	RMSE	0.031
			MAPE	0.068
		LSTM	RMSE	0.026
			MAPE	0.033
Load Modeling [39,88,96]	68-bus New England and New York Interconnect System	ReLU	RMSE	0.045
			MAPE	0.074
		SAE	RMSE	0.041
			MAPE	0.069
		CNN	RMSE	0.036
			MAPE	0.049
		LSTM	RMSE	0.029
			MAPE	0.032
Demand Forecasting [40,97–103]	Household Electric Power Consumption	CNN	RMSE	0.083
			MAPE	0.096
		LSTM	RMSE	0.075
			MAPE	0.084
		CNN-LSTM	RMSE	0.048
			MAPE	0.055
Wind Energy Forecasting [33,49,89,104,105]	Wind Integration National Dataset	ReLU	RMSE	0.078
			MAPE	0.092
		SAE	RMSE	0.066
			MAPE	0.084
		CNN	RMSE	0.059
			MAPE	0.079
		LSTM	RMSE	0.045
			MAPE	0.068
		CNN-LSTM	RMSE	0.029
			MAPE	0.036
Solar Energy Forecasting [90,106,107]	Solar Power Data for Integration Studies	ReLU	RMSE	0.093
			MAPE	0.104
		SAE	RMSE	0.085
			MAPE	0.093
		CNN	RMSE	0.071
			MAPE	0.083
		LSTM	RMSE	0.059
			MAPE	0.074
		CNN-LSTM	RMSE	0.034
			MAPE	0.056

Table 1. *Cont.*

Application	Dataset	Model	Performance Metric	Result
Non-intrusive Load Monitoring [15,16,108–110]	Reference Energy Disaggregation Dataset	ReLU	Precision Recall F-score	80.54 59.92 68.72
		SAE	Precision Recall F-score	82.38 61.09 70.16
		CNN	Precision Recall F-score	87.24 63.39 73.43
		LSTM	Precision Recall F-score	89.83 65.78 75.93
		CNN-LSTM	Precision Recall F-score	92.26 68.14 78.39
PMU Event Classification [111–114]	IEEE 34-bus system	CNN	Precision Recall F-score	87.13 71.04 78.27
		CNN-LSTM	Precision Recall F-score	89.15 73.36 80.49
Power Fluctuation Identification [43]	Market Trading Reports	ReLU	MAE MAPE	0.042 1.079
		LSTM	MAE MAPE	0.038 1.057
Fault Identification [91,115–118]	New England 39-bus test system	ReLU	Precision Recall F-score	86.23 70.36 77.49
		SAE	Precision Recall F-score	88.14 72.26 79.41
		CNN	Precision Recall F-score	89.97 75.50 82.10
		LSTM	Precision Recall F-score	91.72 77.18 83.82

2.2. Stacked Autoencoder

Stacked autoencoder (SAE) neural networks are a class of deep neural networks that use unsupervised learning to pretrain multiple layers of the network before fine-tuning the entire network with supervised learning [45]. Autoencoders are a type of neural network that learns to compress and reconstruct the input data. Stacked autoencoders use multiple layers of autoencoders to learn increasingly complex representations of the input data. The neural architecture of a stacked autoencoder consists of multiple autoencoders where each autoencoding building block consists of an encoder network and a decoder network. The encoder network maps the input data to a compressed representation, while the decoder network maps the compressed representation back to the original input data. The objective of the autoencoder is to minimize the reconstruction error between the input and the output, which is typically measured using a loss function such as mean squared error (MSE) [24,119].

The first layer of the stacked autoencoder is trained as a standard autoencoder, using the input data as both the input and the target output. The compressed representation learned by this layer is then used as the input to the next layer of the network. This process is repeated for each subsequent layer of the network, with each layer learning to compress and reconstruct the output of the previous layer. Once all the layers have been pretrained in an unsupervised manner, the entire network is fine-tuned using supervised learning to learn the final classification or regression task. This involves adding one or more output layers to the network, and updating all the weights of the network using backpropagation and a supervised loss function such as cross-entropy or mean squared error.

The pretraining phase of the stacked autoencoder allows the network to learn useful highly nonlinear feature representations of the input data in an unsupervised manner. This can be particularly useful for tasks with limited labeled data [120], as the network can learn to extract relevant features from the unlabeled data to improve its performance on the labeled data. Additionally, the use of multiple layers in the stacked autoencoder allows for the learning of hierarchical representations of the data, with each layer learning to extract increasingly abstract and complex features [24].

The mathematical equations for the pretraining phase of the stacked autoencoder can be written as follows. Let x be the input data, and f_i and g_i be the encoder and decoder functions for the i-th layer of the network (i.e., i-th autoencoder), respectively. The compressed representation h_i learned by the i-th layer is given by $h_i = f_i(h_{i-1})$ where $h_0 = x$. The reconstructed output of the i-th layer is computed via $\hat{h}_{i-1} = g_i(h_i) \simeq h_{i-1}$. The objective of the i-th autoencoder is to minimize the reconstruction error between its input h_{i-1} and the reconstructed output $\hat{h}_{i-1}$, which is typically measured using mean squared error $L_i = ||h_{i-1} - \hat{h}_{i-1}||_2^2$. The pretraining phase involves minimizing the reconstruction error for each layer of the network, using the compressed representation h_i learned by the previous layer as the input. The weights of each layer are updated using backpropagation through the decoder network, with the gradients of the reconstruction error with respect to the weights being backpropagated through the decoder network and then through the encoder network. Once all the layers have been pretrained in this manner, the entire network is fine-tuned using supervised learning, by adding one or more output layers and updating all the weights of the network using backpropagation and a supervised loss function.

As shown in Table 1, SAEs are widely employed in power systems studies due to their simple implementation and strong unsupervised features captured by these methods. Generally speaking, SAEs provide better classification and regression accuracies in comparison to the deep ReLU model as these techniques seek to learn an unsupervised set of features from the data before mapping the input measurements to the supervised (desired) output. Table 1 shows that SAE yields an RMSE of 0.071 and MAPE of 0.086 for voltage stability assessments of the New England system [92]. Moreover, as shown in the table, SAE improves the accuracy of deep ReLU network in power system reliability evaluation and load modeling, as well as sustainable energy prediction. For instance, the SAE yields 28.8% better RMSE and 14.5% better MAPE compared to the deep ReLU network in the reliability assessment of the IEEE-RTS-79 system [87]. Furthermore, this neural network yields an RMSE of 0.085 and MAPE of 0.093 for the hourly prediction of photovoltaic energy using the Solar Power Data for Integration Studies [107].

2.3. Long Short-Term Memory Network

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that are designed to handle the vanishing and exploding gradient problems that can occur in traditional RNNs [121,122]. LSTMs use a combination of memory cells and gating mechanisms to selectively remember or forget information from previous time steps, allowing them to capture long-term dependencies in sequential data. At each time step t, an LSTM cell receives an input x_t and a hidden state h_{t-1} from the previous time step, and produces an output y_t and a new hidden state h_t. The LSTM cell consists of several gates that control the flow of information into and out of the cell. These gates include the input gate i_t, the forget gate f_t, and the output gate o_t. The input gate determines how much of the input x_t should be added to the cell state c_t. It takes as input the current input x_t

and the previous hidden state h_{t-1} and produces an activation a_i^t that is passed through a sigmoid activation function to produce the gate output i_t:

$$\begin{aligned} a_i^t &= W_{ix}x_t + W_{ih}h_{t-1} + b_i \\ i_t &= \sigma(a_i^t) \end{aligned} \tag{1}$$

The forget gate determines how much of the previous cell state c_{t-1} should be retained in the current cell state c_t. It takes as input the current input x_t and the previous hidden state h_{t-1}, and produces an activation a_f^t that is passed through a sigmoid activation function to produce the gate output f_t:

$$\begin{aligned} a_f^t &= W_{fx}x_t + W_{fh}h_{t-1} + b_f \\ f_t &= \sigma(a_f^t) \end{aligned} \tag{2}$$

The cell state c_t is updated by:

$$\begin{aligned} \tilde{c_t} &= \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \\ c_t &= f_t * c_{t-1} + i_t * \tilde{c_t} \end{aligned} \tag{3}$$

The output gate determines how much of the cell state c_t should be used to compute the output y_t. It takes as input the current input x_t and the previous hidden state h_{t-1} and produces an activation a_o^t that is passed through a sigmoid activation function to produce the gate output o_t:

$$\begin{aligned} a_o^t &= W_{ox}x_t + W_{oh}h_{t-1} + b_o \\ o_t &= \sigma(a_o^t) \end{aligned} \tag{4}$$

The final output y_t is computed by multiplying the cell state c_t by the output gate o_t and passing it through a Tanh activation function:

$$y_t = \tanh(c_t) * o_t \tag{5}$$

LSTM networks can be stacked to create deeper architectures, with each layer consisting of multiple LSTM cells [123]. The hidden state of the previous layer is passed as input to the current layer, allowing the network to learn increasingly abstract and complex representations of the input data [24].

Due to their high generalization power and great adaptation to new time-dependent patterns and scenarios, LSTMs are widely used in power system research, especially for temporal pattern recognition in time series datasets and measurements. For instance, as shown in Table 1, LSTM shows high accuracy with an RMSE of 0.035 and MAPE of 0.046 for the voltage stability assessment in the New England 10-generator test system [93]. Furthermore, this model shows the best performance among discriminative neural architectures with an RMSE of 0.026 and MAPE of 0.033 for power system reliability assessment in the IEEE-RTS-79 system [94]. The LSTM units are also employed for load modeling and have shown state-of-the-art performance with an RMSE of 0.029 and MAPE equal to 0.032 for load parameter identification in the 68-bus New England and New York Interconnect power grid [39]. In addition, the time series pattern recognition tasks such as load forecasting, wind prediction, and solar power prediction are time-dependent tasks where LSTM has shown state-of-the-art performance [99]. For instance, in the hourly load prediction of the Household Electric Power Consumption dataset, the LSTM network shows great performance with 0.075 RMSE and 0.084 MAPE. Various classification tasks can also be accurately solved through LSTM units. For example, in energy disaggregation, the LSTM network shows a precision of 89.83 and F-score of 75.93, which is close to the state-of-the-art performance [108]. Additionally, in fault identification, the LSTM network shows the best results with a precision of 91.72 and recall of 77.18 which gives a classification F-score of 83.82 [115].

In addition to recurrent structures, the following non-recursive techniques can be applied to capture temporal dependencies in the data:

(1) Feature Engineering: By creating relevant features that represent the temporal characteristics of the data, one can capture time-dependent patterns indirectly. For instance, we can compute statistical features such as rolling means, standard deviations, or exponential moving averages over different time windows. These features can provide insights into trends, seasonality, or other temporal patterns and have been applied to battery state-of-charge estimation [124] and load forecasting [125].

(2) Time Encoding: One can encode the timestamp or time information into a format that can be easily interpreted by machine learning algorithms. For example, we can convert timestamps into cyclical representations such as hour of the day, day of the week, or month of the year. This encoding can help algorithms capture periodic patterns and time dependencies and has been applied to energy disaggregation problems [45].

(3) Nonlinear Transformations: Sometimes nonlinear transformations of the data can reveal time-dependent structures that are not evident in the original form. For instance, we can apply mathematical functions such as logarithmic or exponential transformations to the data. These transformations can help normalize the data or highlight specific temporal patterns. This method has been applied to power system state estimation [126], fault detection [127], and power grid time domain simulation [128].

(4) Dynamic Time Warping (DTW): DTW is a technique used to measure the similarity between two time series, even when they have different lengths or shapes. This method allows for finding the optimal alignment between two sequences, which can help capture time dependencies. DTW can be useful when we have similar patterns occurring at different time scales or with time shifts. This technique is widely applied to peak load prediction [129], fault diagnosis [130], and controlled islanding [131].

2.4. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of neural network that is specifically designed for the classification and regression of vectors as well as two-dimensional data points. CNNs use convolutional layers to extract features from vectors and matrices, and pooling layers to reduce the spatial dimensions of the feature maps [95,96,132]. The final layers of the CNN consist of fully connected layers that perform the classification or regression task. The key components of a CNN are the convolutional layers, which perform the convolution operation on the input data using a set of learnable filters. The output of the convolution operation is a feature map that represents the activation of the filter at each location of the input data. The filter weights are learned during the training process via backpropagation [101].

The convolution operation can be represented mathematically by:

$$z_{ij} = \sum_{p=0}^{k-1} \sum_{q=0}^{k-1} x_{i+p,j+q} w_{p,q} + b \tag{6}$$

where $x_{i,j}$ is the pixel value at location (i, j) of the input data, $w_{p,q}$ is the weight of the filter at position (p, q), b is the bias term, and k is the size of the filter. The output feature map h_{ij} is obtained by passing the result of the convolution operation through a nonlinear activation function, such as the rectified linear unit (ReLU) function [96]. After each convolutional layer, a pooling layer is typically added to reduce the spatial dimensions of the feature maps. The most common type of pooling layer is the max pooling layer, which takes the maximum value within a local region of the feature map. The max pooling operation can be represented mathematically as:

$$h_{ij} = \max_{p,q} h_{ip,jq} \tag{7}$$

The final layers of the CNN consist of fully connected layers, which perform the supervised task. The output of the last convolutional layer is flattened into a vector and passed through a series of fully connected layers, each of which applies a linear transformation to the input followed by a nonlinear activation function [115]. The final output of the network is a probability distribution over the different classes in classification or the probability of real-valued numbers in regression.

As shown in Table 1, CNNs are widely applied for various applications in power systems due to their high generalization power, feature sparsity, and the spatial and temporal coherence assumption of this model that usually holds in most real-world input measurements. For instance, CNN is combined with LSTM networks for the assessment of voltage stability for the New England 10-generator 39-bus system, which results in an RMSE of 0.021 and a MAPE of 0.038 [28]. This is the state-of-the-art performance compared with other deep neural architectures including SAEs and LSTMs. The main reason for such accuracy improvements is the fact that CNN is a cutting-edge solution for capturing spatial data dependencies, while the LSTM is among the best models for modeling temporal dependencies in power systems measurements [96]. CNNs also show great performance in the reliability assessment of power grids. These methods improve the results of the deep ReLU network by 31.1% RMSE and 18.1% MAPE on the IEEE-RTS-79 system [95]. Similar improvements are shown in load modeling and demand forecasting [98]. In sustainable energy prediction, the CNN provides higher accuracy in comparison to deep ReLU networks and SAEs since the CNN is capable of capturing the spatial dependencies between the wind or solar power measurements of neighboring sites while the classing deep ReLU and SAE-based techniques can merely handle one station at a time and do not have enough capacity to see the entire stations in a wide region. Moreover, the sparsity of CNN features helps this model to need less number of training samples to achieve a high accuracy. The CNN also improves the energy disaggregation accuracy of SAE by 5.9% precision and 3.76% recall on the Reference Energy Disaggregation Dataset [15]. This model also shows state-of-the-art classification performance for PMU event categorization of the IEEE 34-bus system with a precision of 89.15 and recall of 73.36 when it is incorporated into the LSTM units [112]. Similar high accuracies can be seen in the fault identification studies where the CNN provides a high F-score equal to 82.10 for the New England 39-bus system [116].

2.5. Strengths and Shortcomings of Discriminative Deep Architectures

Figure 1 outlines the strengths and weaknesses of various deep discriminative models within the context of power system research. As detailed in the table, the ReLU neural network offers a simple implementation characterized by low training time complexity and fast feedforward mechanisms. Nevertheless, this supervised model does not inherently account for temporal or spatial features within datasets. ReLU networks lack explicit mechanisms for capturing spatial information in data. They are primarily designed for processing vector inputs, where the spatial relationships between the input features are not considered. That is, the ReLU operations do not assume any similarity metrics between their input variables (i.e., vector entries). Therefore, ReLU networks may struggle to effectively handle data with inherent spatial coherence, such as images or sequential data. On the other hand, CNNs are specifically designed to capture spatial coherence in data since they apply convolutional layers that define a set of learnable filters or kernels to local regions of the input data, scanning the entire input through the filters to extract relevant features [24]. The SAE shares a similar limitation as it cannot well capture spatial and temporal patterns directly. However, SAE is suited for unsupervised learning or learning with limited data. Unlike SAE, LSTM can process temporal data due to its recurrent structure, which enables the model to represent time-dependent relationships and variable-sized input. However, due to the larger quantity of parameters in LSTM compared to conventional recurrent neural networks, LSTM runs a higher risk of overfitting and shows significant sensitivity to observation noise. Finally, CNN leverages filtering and pooling layers to extract informative sparse representations from spatial datasets using straightforward gradient-based methods. Given that the filters can be trained in a distributed fashion, CNN emerges as a highly efficient methodology for pattern recognition in large-scale systems. However, due to the lack of a recursive structure in this supervised model, it struggles to accurately capture time-dependent structures of the data.

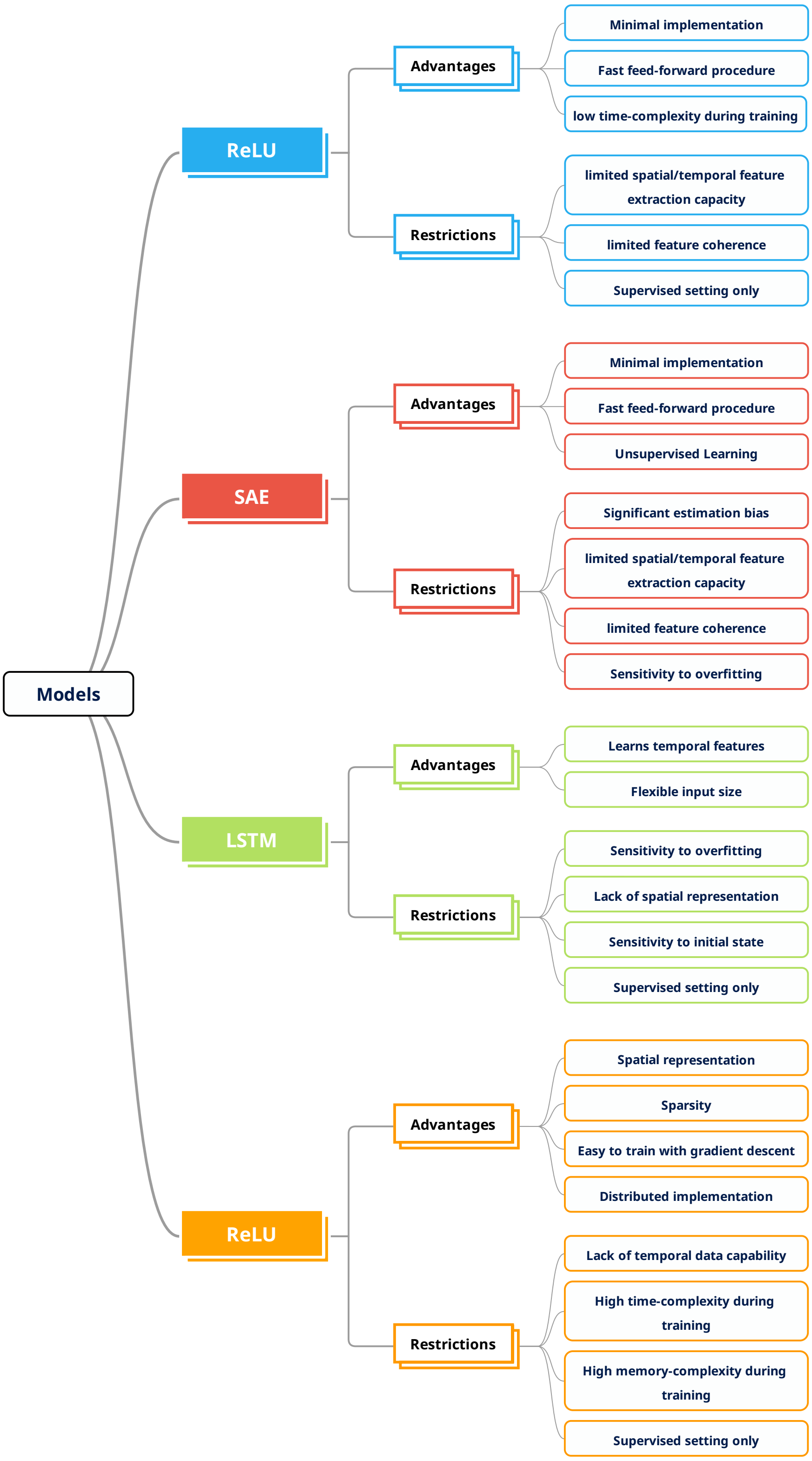
Models
ReLU
Advantages
Minimal implementation
Fast feed-forward procedure
low time-complexity during training
Restrictions
limited spatial/temporal feature extraction capacity
limited feature coherence
Supervised setting only
SAE
Advantages
Minimal implementation
Fast feed-forward procedure
Unsupervised Learning
Restrictions
Significant estimation bias
limited spatial/temporal feature extraction capacity
limited feature coherence
Sensitivity to overfitting
LSTM
Advantages
Learns temporal features
Flexible input size
Restrictions
Sensitivity to overfitting
Lack of spatial representation
Sensitivity to initial state
Supervised setting only
ReLU
Advantages
Spatial representation
Sparsity
Easy to train with gradient descent
Distributed implementation
Restrictions
Lack of temporal data capability
High time-complexity during training
High memory-complexity during training
Supervised setting only

Figure 1. Strengths and Shortcomings of Discriminative Deep Architectures.

3. Probability-Based Deep Neural Architectures

Probability-based Deep Neural Architectures, also known as Generative Neural Networks (GNNs), are a class of neural networks that are capable of generating new data samples that resemble the input data [132,133]. The GNNs can also learn the probability distribution of the input data. This means that the GNN can generate new samples that not only resemble the input data but also follow the same statistical patterns as the input data. Therefore, this class of neural networks is widely used for probabilistic estimations such as probabilistic load modeling [134], uncertainty-aware sustainable energy prediction [135], and power grid synthesis [136].

3.1. Deep Belief Network

Deep Belief Networks (DBNs) are a class of neural networks that are composed of multiple layers of Restricted Boltzmann Machines (RBMs). RBMs are generative models that consist of two layers: a visible layer and a hidden layer. The visible layer represents the input data, and the hidden layer represents the latent variables that capture the underlying structure of the input data. RBMs are trained using the Contrastive Divergence algorithm [137], which maximizes the log-likelihood of the training data.

DBNs are trained using a greedy layer-wise approach, where each layer is trained independently using an unsupervised learning algorithm such as the Contrastive Divergence algorithm. After each layer is trained, the output of the previous layer is used as the input to the current layer. Once all the layers are trained, the DBN can be fine-tuned using a supervised learning algorithm such as backpropagation.

The probability of the input data given the hidden variables in an RBM is defined by the energy function:

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}v_ih_j - \sum_{i=1}^{V} a_iv_i - \sum_{j=1}^{H} b_jh_j \tag{8}$$

where $\boldsymbol{v}$ is the visible layer, $\boldsymbol{h}$ is the hidden layer, V is the number of visible units, H is the number of hidden units, w_{ij} is the weight between visible unit i and hidden unit j, a_i is the bias of visible unit i, and b_j is the bias of hidden unit j.

The joint probability distribution of the visible layer and hidden layer is given by the Boltzmann distribution:

$$p(\boldsymbol{v},\boldsymbol{h}) = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{Z} \tag{9}$$

where Z is the partition function, which normalizes the distribution. The conditional probability of the hidden layer given the visible layer is given by:

$$p(\boldsymbol{h}|\boldsymbol{v}) = \prod_{j=1}^{H} p(h_j|\boldsymbol{v}) = \prod_{j=1}^{H} \sigma(\sum_{i=1}^{V} w_{ij}v_i + b_j) \tag{10}$$

where σ is the sigmoid function. The conditional probability of the visible layer given the hidden layer is given by:

$$p(\boldsymbol{v}|\boldsymbol{h}) = \prod_{i=1}^{V} p(v_i|\boldsymbol{h}) = \prod_{i=1}^{V} \sigma(\sum_{j=1}^{H} w_{ij}h_j + a_i) \tag{11}$$

where σ is the sigmoid function.

DBNs are trained using a greedy layer-wise approach, where each layer is trained independently using the Contrastive Divergence. This training process is composed of two phases: pretraining and fine-tuning.

In the pretraining phase, each layer is trained independently as an RBM. The input to each RBM is the output of the previous layer, or the raw input data in the case of the first layer. The Contrastive Divergence algorithm is used to maximize the log-likelihood of the training data, by adjusting the weights and biases of the RBM. The goal of pretraining is to learn a set of feature detectors that can capture the underlying structure of the input data. Once all the layers have been pretrained, the DBN is fine-tuned using a supervised learning algorithm such as backpropagation. During the fine-tuning phase, the weights and biases of the entire network are adjusted to minimize a cost function, which is typically the cross-entropy between the predicted labels and the true labels. The output of the DBN is obtained by passing the input data through each layer, and the output of the final layer is used as the prediction.

The advantage of the greedy layer-wise approach is that it allows the DBN to learn a hierarchy of features, where each layer captures increasingly complex and abstract features [24]. This makes the DBN more robust to variations in the input data, and enables it to generalize well to new, unseen data [57]. However, the disadvantage of the greedy layer-wise approach is that it may not always lead to the optimal solution, and the pretraining phase may take a long time to converge. During pretraining, the convergence of each layer of DBN may be too slow due to the following challenges: (1) Initialization: RBMs are sensitive to weight initialization [57]. Initializing the weights inappropriately can result in slow convergence or even convergence to poor local optima. Proper initialization techniques, such as sampling from a small Gaussian distribution or using unsupervised pre-training, can help accelerate convergence; (2) Learning Rate: The learning rate determines the step size at each update during training. If the learning rate is too high, it can cause unstable updates and hinder convergence. On the other hand, if the learning rate is too low, the convergence may become excessively slow. Finding an appropriate learning rate or using adaptive learning rate methods can be crucial for efficient convergence [25]; (3) Sampling Method: RBMs use sampling techniques, such as Gibbs sampling or Contrastive Divergence, to approximate the model's distribution and update the weights. The number of sampling steps can significantly impact convergence. Using a higher number of sampling steps improves the approximation but increases the computational cost, potentially slowing down convergence [58,59]; and (4) Local Optima: Similar to many optimization problems, RBMs can get stuck in local optima, where the network fails to find the global minimum of the objective function. The network may converge to a suboptimal solution, leading to slower convergence or reduced performance. Exploring different weight initializations, employing techniques such as momentum or simulated annealing, or using more advanced optimization algorithms can help overcome local optima and improve convergence [24].

DBNs are widely applied to various power systems problems in recent studies. Table 2 shows the applications of the probabilistic deep neural architectures in power systems research. As shown in this table, DBN is applied to the transient stability assessment problem of the Central China Region Power Grid and has shown a precision of 80.24, recall of 78.92, and an F-score equal to 79.57 in this application [138]. Moreover, DBN is shown to accurately learn probabilistic features from time-dependent datasets [57]. For instance, DBN provides highly accurate hourly demand predictions on the Texas Urbanized Area Dataset with an RMSE and MAPE of 0.045 and 0.091, respectively [139]. Additionally, this model has shown great prediction accuracies for hourly sustainable power prediction tasks. For example, on the Solar Integration National Dataset, the DBN shows a prediction RMSE of 0.082 and MAPE equal to 0.093 [140]. State estimation is another major application of DBNs in power engineering research. This deep learning framework yields an RMSE and MAPE of 0.092 and 0.156, respectively, on the US PG&e69 power system [11]. Moreover, DBN is employed for classification tasks such as fault identification as well as cyberattack detection. In fault classification, DBN provides a classification precision, recall, and F-score of 81.32, 76.59, 78.88, respectively, when applied to the IEEE 33-bus distribution network [13]. In cyberattack categorization, DBN yields an F-score of 79.44, which shows the great generalization capacity and highly nonlinear decision boundary of this type of deep learning technique in classification tasks [141]. Similar high accuracies are reported for data synthesis problems. For instance, this model shows an RMSE of 0.085 and MAPE of 0.125 in the synthesis of the Columbia University Synthetic Power Grid [121].

Furthermore, it offers an F-score of 74.75 for the disaggregation of residential loads in the Reference Energy Disaggregation Dataset [45].

Table 2. Generative deep neural networks in power system studies.

Application	Dataset	Model	Performance Metric	Result
Transient Stability Assessment [25,58,138,142–144]	Central China Regional Power Grid	DBN	Precision Recall F-score	80.24 78.92 79.57
		VAE	Precision Recall F-score	85.64 81.05 83.28
		GAN	Precision Recall F-score	87.32 84.12 85.69
Demand Forecasting [139,145–149]	Texas Urbanized Area Dataset	DBN	RMSE MAPE	0.045 0.091
		VAE	RMSE MAPE	0.036 0.075
		GAN	RMSE MAPE	0.028 0.066
Wind Power Prediction [5–7,150,151]	Wind Integration Dataset	DBN	RMSE MAPE	0.078 0.095
		VAE	RMSE MAPE	0.064 0.071
		GAN	RMSE MAPE	0.051 0.063
Solar Energy Prediction [8–10,140,152–156]	Solar Integration National Dataset	DBN	RMSE MAPE	0.082 0.093
		VAE	RMSE MAPE	0.071 0.082
		GAN	RMSE MAPE	0.062 0.070
State Estimation [11,12,157]	US PG&E69 distribution system	DBN	RMSE MAPE	0.092 0.156
		VAE	RMSE MAPE	0.074 0.093
		GAN	RMSE MAPE	0.065 0.081
Fault Identification [13,158–166]	IEEE-33 node 10kV distribution network	DBN	Precision Recall F-score	81.32 76.59 78.88
		VAE	Precision Recall F-score	87.45 79.69 83.39
		GAN	Precision Recall F-score	90.05 85.13 87.52

Table 2. *Cont.*

Application	Dataset	Model	Performance Metric	Result
Cyberattack Identification [141,167–171]	Industrial control system cyber attack datasets	DBN	Precision Recall F-score	85.06 74.52 79.44
		VAE	Precision Recall F-score	89.19 76.31 89.25
		GAN	Precision Recall F-score	92.28 78.56 82.25
Power Grid Synthesis [65,172–176]	Columbia University Synthetic Power Grid	DBN	RMSE MAPE	0.085 0.125
		VAE	RMSE MAPE	0.052 0.096
		GAN	RMSE MAPE	0.033 0.071
Energy Disaggregation [177–180]	Reference Energy Disaggregation Dataset	DBN	Precision Recall F-score	78.64 71.23 74.75
		VAE	Precision Recall F-score	85.34 82.95 84.13
		GAN	Precision Recall F-score	92.05 89.73 90.88

3.2. Variational Autoencoder

Variational Autoencoders (VAEs) are a class of generative neural networks that use an encoder and a decoder to learn a low-dimensional representation of high-dimensional data. VAEs are commonly used for image and video generation, but can also be applied to other types of data, such as text and audio. The VAE architecture consists of two neural networks: an encoder network, $q_\phi(z|x)$, which maps the input data, x, to a low-dimensional latent representation, z, and a decoder network, $p_\theta(x|z)$, which maps the latent representation, z, back to the original data space. The objective of the VAE is to learn the joint probability distribution of the input data and the latent representation, $p(x,z)$, and use it to generate new samples. The encoder network maps the input data, x, to a latent representation, z, using a Gaussian distribution with mean, μ, and standard deviation, σ, which are functions of the input data, x. Hence, we have:

$$\begin{aligned} \mu &= f_{\phi,\mu}(x) \\ \log\sigma^2 &= f_{\phi,\sigma}(x) \\ z &\sim \mathcal{N}(\mu,\sigma^2 I) \end{aligned} \tag{12}$$

where $f_{\phi,\mu}$ and $f_{\phi,\sigma}$ are the encoder network functions, and $\mathcal{N}(\mu,\sigma^2 I)$ is the Gaussian distribution with mean μ and covariance matrix $\sigma^2 I$. The standard deviation is usually modeled using the logarithm to ensure positivity. The decoder network maps the latent representation, z, back to the original data space, x, using a similar Gaussian distribution with mean, μ', and standard deviation, σ':

$$\begin{aligned} \mu' &= g_{\theta,\mu}(z) \\ \log\sigma'^2 &= g_{\theta,\sigma}(z) \\ x &\sim \mathcal{N}(\mu',\sigma'^2 I) \end{aligned} \tag{13}$$

where $g_{\theta,\mu}$ and $g_{\theta,\sigma}$ are the decoder network functions. During the training process, the VAE learns to maximize the evidence lower bound (ELBO), which is the lower bound on the log-likelihood of the data under the model [181]. The ELBO is computed by:

$$\begin{aligned} \text{ELBO}(x) &= \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\text{KL}}(q_\phi(z|x)||p(z)) \\ &= \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - \text{KL}(q_\phi(z|x)||p(z)) \end{aligned} \tag{14}$$

where D_{KL} is the Kullback–Leibler divergence [158] between the encoder distribution and the prior distribution, $p(z)$, which is usually assumed to be a unit Gaussian distribution. The first term in the ELBO is the reconstruction loss, which measures the similarity between the input data and the reconstructed data, while the second term is the regularization term, which encourages the latent representation to follow the prior distribution [64].

During the training process, the weights of the encoder and decoder networks are updated using backpropagation and stochastic gradient descent, where the gradient is backpropagated from the ELBO objective function to the network weights [50]. As the VAEs learn the probability distribution of the input data, they can be utilized to generate synthetic samples that follow the distribution of the original data. To generate a data sample $\tilde{x} \sim P(x|z)$, one can generate a random Gaussian sample $\tilde{z}$ with the same dimension as the VAE's latent feature from the Normal distribution, and feed $\tilde{z}$ into the decoder ANN that implements $p_\theta(x|z)$ to generate a synthetic data sample in its output layer.

The VAEs have been applied to a wide range of power systems applications as both an unsupervised feature extraction model and a supervised technique for supervised estimation of discrete labels (classification) or continuous variables (regression). As shown in Table 2, in this area, VAE is applied to the transient stability assessment of the Central China Regional Power Grid, which results in a precision, recall, and F-score of 85.64, 81.05, and 83.28, respectively [143]. These results show a significant improvement over the DBN method since is better able to model the uncertainties in the data using its variational loss function compared to the DBN which is more data-hungry due to its Gibbs sampling-based training process. Additionally, VAE is applied to a wide range of time series prediction tasks such as hourly prediction of electricity demand in residential units, wind energy, and photovoltaic power [182]. VAE provides a slightly better demand prediction model compared to the DBN when applied to the Texas Urbanized Area Dataset for hourly prediction of residential loads [145]. This model results in RMSE and MAPE of 0.036 and 0.075, respectively, which are slightly lower than the DBN. In addition, the results of VAE's hourly wind power predictions on the Wind Integration Dataset show an RMSE and MAPE of 0.064 and 0.071, respectively [150]. The RMSE of VAE in this application is 17.94% and its MAPE is 25.26% better than the DBN method as it can provide a more reliable estimation of the probability densities of the underlying data. Similar improvements can also be seen in the state estimation applications where the VAE provides an RMSE of 0.074 and MAPE of 0.093 on the US PG&E69 distribution network [11]. In terms of classification problems, the VAE outperforms the DBN as it is more data-efficient and robust to noise and uncertainties. For instance, the VAE yields an F-score of 83.39 on the IEEE 33-bus system which is 5.72% higher than the DBN in fault identification [158]. Similar improvements can be seen in attack recognition problems, which are classification challenges [141,170]. VAEs can also be employed for data augmentation and synthesis of power networks. In this domain, they provide an RMSE and MAPE of 0.052 and 0.096, respectively, for the synthesis of the Columbia University Synthesis Power Grid [136]. Additionally, the VAEs outperform DBNs in non-intrusive load monitoring and energy disaggregation tasks. For instance, on the Reference Energy Disaggregation Dataset, VAE provides a 12.55% better load decomposition F-score compared with a similar technique using DBN for feature extraction [177].

3.3. Generative Adversarial Network

Generative Adversarial Networks (GANs) are a class of generative neural networks that use two neural networks, a generator and a discriminator, to learn the underlying probability distribution of the data [133]. The GAN architecture consists of two neural networks: a generator network, G, and a discriminator network, D. The generator network takes a random noise vector, z, as input, and produces a sample, x, from the target distribution,

$p(x)$. The discriminator network takes a sample, x, as input, and produces a binary output, d, indicating whether the sample is real or fake. The objective of the generator network is to produce samples that are indistinguishable from real samples, while the objective of the discriminator network is to correctly distinguish between real and fake samples [161].

The training process of GANs can be formulated as a minimax game between the generator network and the discriminator network [160]. The generator network tries to minimize the difference between the distribution of its generated samples, $G(z)$, and the distribution of real samples, $p(x)$, while the discriminator network tries to maximize the difference between the distribution of real samples and the distribution of fake samples, $D(G(z))$. The objective function for the GAN can be written as:

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{15}$$

where $p_{\text{data}}(x)$ is the true data distribution and $p_z(z)$ is the noise distribution. The first term in the objective function maximizes the log-likelihood of the discriminator network correctly classifying real samples, while the second term maximizes the log-likelihood of the discriminator network incorrectly classifying fake samples [175]. During the training process, the generator network tries to generate samples that fool the discriminator network, while the discriminator network tries to correctly classify real and fake samples. The weights of the generator network and the discriminator network are updated using backpropagation and stochastic gradient descent, where the gradient is backpropagated from the discriminator network to the generator network.

The advantage of GANs is that they can generate realistic samples that capture the underlying distribution of the training data. However, the training of GANs can be difficult and unstable, and the quality of the generated samples may depend heavily on the architecture and hyperparameters of the network. In addition, GANs may suffer from mode collapse [155], where the generator network only learns to generate a few modes of the target distribution, rather than capturing the entire distribution.

Table 2 shows the applications of GANs in power systems research and clear comparisons with VAEs. GANs offer several advantages over VAEs. Firstly, GANs produce more realistic and high-quality synthetic samples compared to VAEs. GANs leverage a discriminator network that learns to distinguish between real and generated samples, leading to sharper and more convincing output. Secondly, GANs do not suffer from the blurry and noisy output issue commonly observed in VAEs. VAEs tend to average over the latent space, resulting in less distinct and fuzzy synthetic data samples. In contrast, GANs generate samples by directly modeling the data distribution, allowing for more diverse and detailed outputs. For instance, as shown in Table 2, GAN shows 2.89% better F-score compared to VAE for the transient stability assessment of the Central China Regional Power Grid [144]. Additionally, GAN improves the MAPE of demand forecasting on the Texas Urbanized Area Dataset by 12.0% compared to the VAE model [148]. Similar improvements can be seen in other time series forecasting tasks in the power system area. For example, as shown in the table, GAN provides a 14.63% less MAPE compared to the VAE for the hourly prediction of solar energy using the Solar Integration National Dataset [154]. In state estimation of the US PG&E69 distribution power network, GAN yields an RMSE of 0.065, which is 12.16% lower than the same model that employs a VAE [12]. Furthermore, for fault identification, GAN provides a 4.95% better F-score in comparison with the VAE on the IEEE 33-bus system [160,161]. Similar improvements can be seen in the cyberattack detection problem as a challenging classification task [170]. As discussed in this section, GANs are capable of generating realistic data points as they accurately learn data probability densities. Therefore, as shown in Table 2, these methods are applied to generate synthetic power networks. On the Columbia University Synthetic Power Grid dataset, the synthesis task by GAN shows an RMSE of 0.033 and MAPE of 0.071 which are significantly lower than applying VAEs and DBNs for the same task to generate realistic power networks [136]. Similar improvements are shown in the table for the energy disaggregation application, where GAN can improve the VAE by 8.02% in the F-score for the disaggregation of residential loads in the Reference Energy Disaggregation Dataset [179].

3.4. Strengths and Shortcomings of Probability-Based Deep Neural Architectures

Figure 2 highlights the benefits and limitations of deep generative modeling as applied to power system research. As the table indicates, DBN, GAN, and VAE exhibit proficiency in managing measurement uncertainties, concurrently offering a robust unsupervised data representation. DBN, in comparison with GAN and VAE, possesses lower sample complexity, which subsequently reduces the amount of training data needed for effective feature extraction. However, DBN's reliance on Gibbs sampling during training introduces considerable time-complexity. Moreover, DBN makes a strong independence assumption regarding its latent variables. Contrarily, GAN and VAE are capable of learning data distributions directly without any prior assumptions, making them well suited for power system data synthesis. Due to its more complex architecture, GAN demands a larger number of training examples relative to DBN. However, GAN's feature diversity is limited, and it does not guarantee parameter convergence. In comparison, while VAE exhibits similar sample complexity to GAN, it provides a more reliable estimation of distribution. Yet, the less pronounced sharpness of VAE relative to GAN makes the latter a superior choice for probabilistic applications.

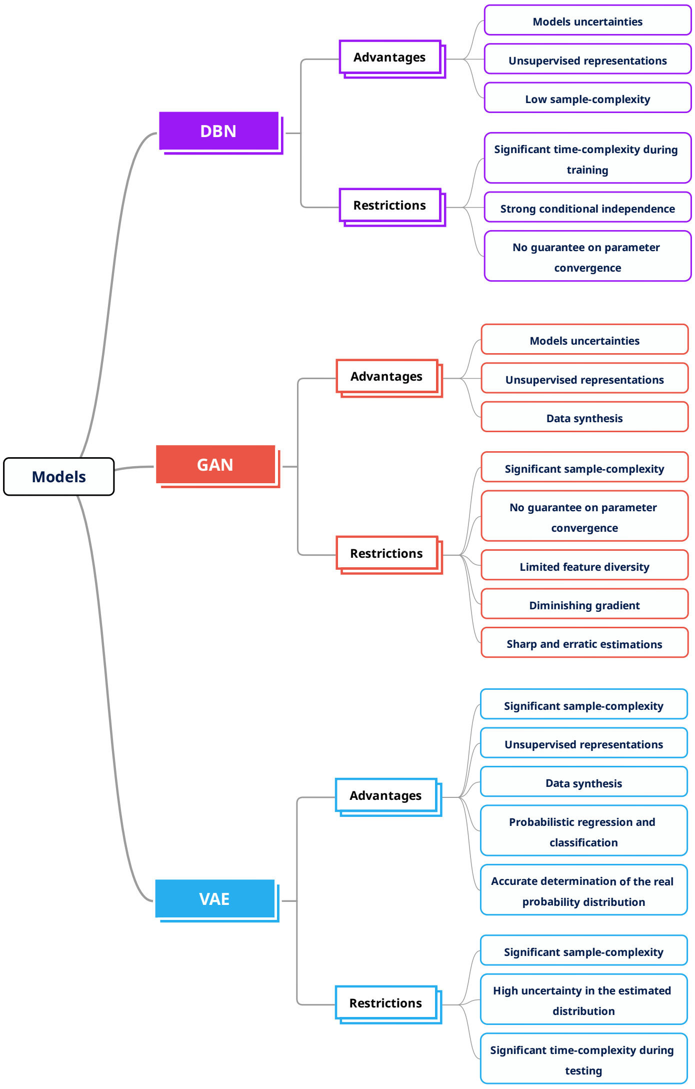
Models
DBN
Advantages
Models uncertainties
Unsupervised representations
Low sample-complexity
Restrictions
Significant time-complexity during training
Strong conditional independence
No guarantee on parameter convergence
GAN
Advantages
Models uncertainties
Unsupervised representations
Data synthesis
Restrictions
Significant sample-complexity
No guarantee on parameter convergence
Limited feature diversity
Diminishing gradient
Sharp and erratic estimations
VAE
Advantages
Significant sample-complexity
Unsupervised representations
Data synthesis
Probabilistic regression and classification
Accurate determination of the real probability distribution
Restrictions
Significant sample-complexity
High uncertainty in the estimated distribution
Significant time-complexity during testing

Figure 2. Strengths and Shortcomings of Deep Generative Models.

4. Deep Reinforcement Learning

Deep reinforcement learning is a subfield of machine learning that deals with training agents to perform tasks in a given environment. In DRL, the agent learns to interact with the environment by taking actions and receiving rewards based on those actions. The goal is to learn a policy that maximizes the expected cumulative reward over time. The key components of a DRL system are the agent, the environment, and the reward function. The agent is typically implemented as a neural network that takes the state of the environment as input and outputs a probability distribution over possible actions. The environment is modeled as a Markov decision process (MDP) which consists of a set of states, actions, transition probabilities, and rewards. The reward function maps states and actions to scalar rewards [183]. The objective of the agent is to learn a policy that maximizes the expected cumulative reward over time. This can be formulated as the expected sum of discounted rewards:

$$J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{16}$$

where θ represents the parameters of the agent's policy, τ represents a trajectory of states and actions, $p_\theta(\tau)$ is the probability of generating a trajectory τ under the agent's policy, r_t is the reward at time step t, and γ is a discount factor that trades off immediate rewards versus future rewards. The key challenge in DRL is to learn an effective policy that can generalize to unseen states and actions. One way to achieve this is to use a neural network to represent the policy [184]. The neural network takes the state of the environment as input and outputs a probability distribution over possible actions. The policy is trained using gradient descent to maximize the expected cumulative reward over time. The gradient of the expected cumulative reward with respect to the policy parameters can be computed using the policy gradient theorem:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}\right] \tag{17}$$

where $\pi_\theta(a_t|s_t)$ is the probability of taking action a_t in state s_t under the policy parameterized by θ. The policy gradient can be estimated using Monte Carlo methods [66] by sampling trajectories from the current policy and computing the gradient of the expected cumulative reward with respect to the policy parameters. To estimate the policy gradient using Monte Carlo, one needs to perform the following steps:

(1) Sample multiple trajectories using the current policy $\pi_\theta(a|s)$. For each trajectory, we compute the returns $G(t) = \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}$ for each time step t using the observed rewards.

(2) Compute the policy gradient estimate by averaging the gradients of the logarithmic policy multiplied by the corresponding returns:

$$\nabla J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T-1} \nabla log(\pi_\theta(a_t^i|s_t^i)) * G(t)^i \tag{18}$$

where N is the number of sampled trajectories and the superscript i denotes the trajectory index.

(3) Update the policy parameters θ using gradient ascent written as $\theta \leftarrow \theta + \alpha * \nabla J(\theta)$ where α is the learning rate.

By repeatedly sampling trajectories, estimating the policy gradient, and updating the policy parameters, the agent can iteratively improve its policy toward maximizing the expected cumulative reward.

4.1. Deep Q-Network (DQN)

Deep Q-Network (DQN) is a deep reinforcement learning algorithm that combines a deep neural network with Q-learning to learn an optimal policy [67,185]. The DQN algorithm has been successfully applied to various tasks, including playing Atari games and controlling robotic systems. The goal of the DQN algorithm is to learn a function $Q(s,a)$ that estimates the expected return for taking action a in state s. The Q-function is updated iteratively using the Bellman equation:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{19}$$

where r is the reward received after taking action a in state s, s' is the next state, α is the learning rate, and γ is the discount factor.

To deal with the curse of dimensionality and to enable the DQN algorithm to learn a function approximation for $Q(s,a)$, a deep neural network is used to represent the Q-function. The network takes a state s as input and outputs Q-values for all possible actions. During training, the DQN algorithm uses experience replay to store the agent's experiences in a replay buffer, which is then used to sample batches of experiences to train the network [46]. The loss function for the DQN is the mean squared error between the Q-value estimated by the network and the target Q-value, which is computed using the Bellman equation:

$$L(\theta) = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q_{target}(s',a';\theta^{-}) - Q(s,a;\theta)\right)^2\right] \tag{20}$$

where θ represents the parameters of the network, $Q_{target}(s',a';\theta^{-})$ is the target Q-value computed using a separate target network with parameters θ^{-}, and the expectation is taken over a batch of experiences.

To stabilize training, the DQN algorithm uses two key techniques: target network [68] and ϵ-greedy exploration [69]. The target network is used to generate the target Q-values and is updated periodically by copying the parameters from the Q-network. The ϵ-greedy exploration strategy is used to encourage the agent to explore the environment by selecting a random action with probability ϵ and the action with the highest Q-value with probability $1-\epsilon$.

Overall, the DQN algorithm is a powerful deep reinforcement learning technique that has been shown to achieve state-of-the-art performance on a wide range of power engineering tasks. Table 3 shows the applications of deep reinforcement learning techniques in power system research. As shown in this table, DQN is a successful technique in the voltage control of the IEEE 123-bus system with an average control reward of 153.46 [186,187]. This method also yields a normalized reward of 0.795 for the emergency control of the IEEE 39-bus system [188]. It is also studied for transportation electricity management using the California Freeway Performance Measurement System dataset, which shows a cost efficiency metric equal to 0.141 [189,190]. In demand-response scheme learning, the DQN provides an operational cost of $161.93 on the Steel Powder Manufacturing Dataset [191,192]. Moreover, the DQN provides a profit of £5.24×10^5 on the ISO New England Inc. environment in electricity market problems [193]. This methodology has also been used to solve power scheduling problems, with an average income of $4268.17 using the Center for Renewable Energy Systems Technology Model [194,195]. Another major application of DQN is the cyberattack detection and identification problem. In this area, the DQN results in precision, recall, and F-score of 83.70, 79.08, and 81.32, respectively, when applied to the IEEE 39-bus system [74,196].

Table 3. Deep Reinforcement Learning neural networks in power system studies.

Application	Dataset	Model	Performance Metric	Result
Voltage Control [66,67,186,187,197–200]	IEEE 123-Bus System	DQN	Average Control Reward	153.46
		DDQN	Average Control Reward	161.74
		DDPG	Average Control Reward	165.92
Emergency Control [188,201–204]	IEEE 39-bus	DQN	Normalized Reward	0.795
		DDQN	Normalized Reward	0.864
		DDPG	Normalized Reward	0.920
Transportation Electrification Management [68,186,189,205–208]	California Freeway Performance Measurement System (PeMS)	DQN	Cost Efficiency (Compared to binary control)	0.141
		DDQN	Cost Efficiency (Compared to binary control)	0.163
		DDPG	Cost Efficiency (Compared to binary control)	0.189
Demand-Response Strategy Learning [70,191,192,209–213]	Steel Powder Manufacturing Dataset	DQN	Operation Cost($)	194.06
		DDQN	Operation Cost($)	175.23
		DDPG	Operation Cost($)	161.93
Electricity Market and Economics [69,193,214–216]	ISO New England Inc.	DQN	Profit (£)	5.24×10^5
		DDQN	Profit (£)	6.74×10^5
		DDPG	Profit (£)	7.34×10^5
Energy Scheduling [194,195,207,217–220]	Centre for Renewable Energy Systems Technology Model	DQN	Average Income($)	$4268.17
		DDQN	Average Income($)	$4805.65
		DDPG	Average Income($)	$5139.06
Cyberattack Detection [14,74,196,221–223]	IEEE 39-Bus System	DQN	Precision Recall F-score	83.70 79.08 81.32
		DDQN	Precision Recall F-score	87.29 91.22 89.21
		DDPG	Precision Recall F-score	94.07 93.38 93.72

4.2. Double DQN

Double Deep Q-Network (DDQN) is an extension of the Deep Q-Network (DQN) algorithm that addresses the problem of overestimation of Q-values. The DDQN algorithm uses two separate deep neural networks to decouple the selection of actions and the evaluation of their Q-values, resulting in more accurate estimates of the Q-function [68]. Similar to DQN, the DDQN algorithm uses experience replay to store the agent's experiences in a replay buffer and uses a deep neural network to represent the Q-function. The Q-function is updated iteratively using the Bellman equation computed by:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s', \arg\max_{a''} Q(s', a''; \theta); \theta^{-}) - Q(s,a) \right] \tag{21}$$

where θ represents the parameters of the Q-network, θ^{-} represents the parameters of a separate target network, r is the reward received after taking action a in state s, s' is the next state, α is the learning rate, and γ is the discount factor.

In DDQN, the Q-values for selecting actions are computed using the Q-network, while the Q-values for evaluating the selected actions are computed using the target network. This decoupling of action selection and evaluation helps to reduce the overestimation of Q-values. The loss function for the DDQN is the mean squared error between the Q-value estimated by the Q-network and the target Q-value, which is computed using the target network as follows:

$$L(\theta) = \mathbb{E}\left[\left(r + \gamma Q(s', \arg\max_{a''} Q(s', a''; \theta); \theta^{-}) - Q(s,a;\theta) \right)^2 \right] \tag{22}$$

where the expectation is taken over a batch of experiences.

Overall, DDQN is a powerful deep reinforcement learning technique that has been shown to achieve state-of-the-art performance on a wide range of tasks while addressing the problem of overestimation of Q-values. As shown in Table 3, the DDQN generally shows better accuracies and objective values compared to the DQN method. For instance, as shown in the table, DDQN provides a 5.39% better average control reward over the DQN for the voltage control of the IEEE 123-bus system [186,187]. Moreover, the DDQN shows 8.68% higher normalized average reward over the DQN in the emergency control task of the IEEE 39-bus test case [201]. Similar results can be seen in the transportation electrification management tasks, where the DDQN results in a 0.163 cost efficiency [186,189]. In the demand-response problems, this method yields a 9.70% lower operational cost compared to the DQN on the Steel Powder Manufacturing model [191,192]. Moreover, in the ISO New England Inc., the DDQN provides a 28.63% higher profit compared to the DQN for electricity market management [193,214]. Similar profit improvements are shown for the energy scheduling of the Center for Renewable Energy Systems Technology model [217,219]. Finally, in the cyberattack identification problem of the IEEE 39-bus system, this method shows a 4.29%, 15.35%, and 9.70% higher precision, recall, and F-score compared with the DQN method [74], which is mainly due to the fact that DDQN handles the problem of overestimation of Q-values that exists in the DQN model.

4.3. Deep Deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient (DDPG) is a model-free, off-policy algorithm for continuous action spaces in reinforcement learning. It combines the ideas of DQN and deterministic policy gradient methods to learn a deterministic policy that maps states to actions.

The DDPG algorithm uses a deep neural network to represent both the actor (policy) and critic (action-value function) networks. The actor network takes in the state as input and outputs the action to be taken. The critic network takes in both the state and action as input and outputs the corresponding action-value function. The Q-value for a state-action pair is defined as:

$$Q(s,a) = \mathbb{E}\Big[r + \gamma Q(s', \mu'(s'); \theta_{Q'})|s,a\Big] \tag{23}$$

where μ is the actor network, $\theta_{Q'}$ are the target network parameters, and μ' is the target actor network that is used to compute the target Q-value. The target Q-value is then used to update the critic network parameters using the Bellman equation:

$$\theta_Q \leftarrow \theta_Q - \alpha_Q \frac{\partial L}{\partial \theta_Q} \tag{24}$$

where L is the mean squared error between the predicted Q-value and the target Q-value.

To update the actor network, the deterministic policy gradient theorem is used. The objective is to maximize the expected return:

$$J(\theta_\mu) = \mathbb{E}_{s_t \sim \rho^\beta, a_t \sim \mu_\theta}[Q^\mu(s_t, a_t)] \tag{25}$$

where ρ^β is a replay buffer and $Q^\mu(s_t, a_t)$ is the Q-value estimated by the critic network for a given state-action pair. The gradient of the objective is given by:

$$\nabla_{\theta_\mu} J(\theta_\mu) \approx \frac{1}{N} \sum_i \nabla_a Q(s,a;\theta_Q)|_{s=s_i, a=\mu(s_i;\theta_\mu)} \nabla_{\theta_\mu} \mu(s;\theta_\mu)|_{s_i} \tag{26}$$

where N is the batch size and $\nabla_a Q(s,a;\theta_Q)$ is the gradient of the Q-value with respect to the action. The actor network is updated by descending along this gradient:

$$\theta_\mu \leftarrow \theta_\mu + \alpha_\mu \nabla_{\theta_\mu} J(\theta_\mu) \tag{27}$$

where α_μ is the learning rate for the actor network.

DDPG has several advantages over deep Q-Network algorithms. Firstly, DDPG is capable of handling continuous action spaces, whereas DQN is primarily designed for discrete action spaces. This makes DDPG well-suited for real-world power engineering tasks. Secondly, DDPG is an off-policy algorithm, meaning it can learn from a batch of experiences collected independently from the current policy. This enables more efficient learning by reusing past experiences and reducing the sample complexity. On the other hand, DQN is an on-policy algorithm that requires interactions with the environment to collect new data for each update. Lastly, DDPG utilizes actor–critic architecture, allowing it to learn both a deterministic policy (actor) and estimate the value function (critic) simultaneously. This improves the stability and convergence of the learning process compared to the separate target and behavior networks used in DQN.

As shown in Table 3, DDPG outperforms DQN and DDQN in various data-driven power engineering studies. For instance, as shown in the table, the DDPG method provides a 2.58% higher average control reward compared to the DDQN for voltage control of the IEEE 123-bus system [67]. DDPG also shows a higher normalized reward in the emergency control problem of the IEEE 39-bus test case compared to the DDQN [203]. It provides a 6.48% better normalized reward value which is mainly due to the fact that DDPG can learn from a batch of experiences and it has better convergence in real-time applications. In addition, DDPG obtains a 15.95% higher cost efficiency in the transportation energy management problem [224] of the California Freeway Performance Measurement System compared to the DDQN method [68,205]. In the demand-response problem of the Steel Powder Manufacturing Dataset, the DDPG shows a 7.59% lower cost compared with the DDQN model [192,209]. A similar improvement can also be observed in the electricity market management case using IOS New England Inc. dataset [214,215]. The average income obtained by the DDPG is 20.40% and 6.94% higher than DQN and DDQN, respectively, in the power scheduling problem solved for the Centre for Renewable Energy Systems Technology model [195]. DDPG also provides better classification accuracies compared with the DQN and DDQN since DDPG shows better stability and convergence during its training and provides a more reliable classification decision boundary. For instance, in cyberattack identification for the IEEE 39-bus test case, DDPG's classification precision, recall, and F-score are 7.76%, 18.08%, and 15.25% higher than DQN, respectively [222,223]. Additionally, DDPG has 5.06% better F-score compared with the DDQN for the same classification problem.

4.4. Strengths and Shortcomings of Deep Reinforcement Learning

Figure 3 shows the strengths and limitations of deep reinforcement learning methodologies in the context of power system implementations. DQN and DDQN exhibit reliable and resilient training dynamics, making them highly applicable to datasets characterized by elevated uncertainty. Nevertheless, these techniques do not ensure the definitive convergence of their parameters. By reducing the overestimation bias of state-action values, Double DQN enables better decision-making and more accurate value estimation, leading to enhanced performance and stability in reinforcement learning tasks. Empirical studies such as [68,225] have shown that Double DQN provides more stable and reliable learning compared with the original DQN algorithm. It exhibits improved convergence properties and can learn more efficiently, especially in domains with high-dimensional state spaces or complex action spaces.

DQN and DDQN primarily concentrate on refining deterministic strategies within discrete action spaces. This implies that, relative to the DDPG, they may not be well suited for practical situations which require continuous actions. The DDPG method has significant benefits for DRL. First, DDPG integrates the advantages of actor-critic methods and deep neural networks, making learning in high-dimensional action spaces efficient and scalable. Second, DDPG utilizes off-policy learning, which enables the agent to learn from a replay buffer of previous experiences, thereby enhancing sample efficiency and overall stability. In addition, DDPG includes target networks that are periodically updated to provide more consistent and reliable value estimations. Despite DDPG potentially being susceptible to parameter instability during its training phase, it guarantees fast and reliable convergence towards an accurate stochastic local policy.

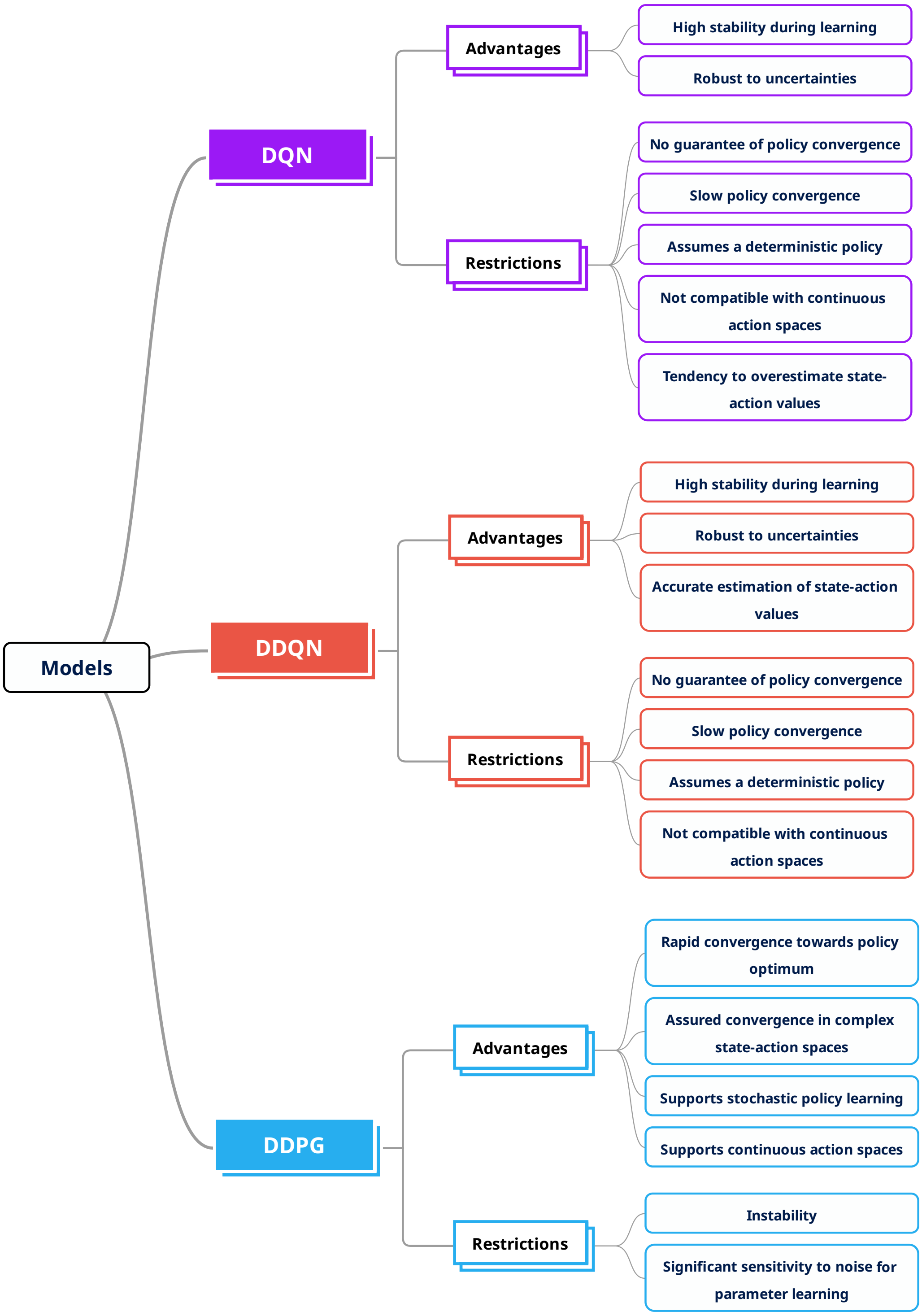
Models
DQN
Advantages
High stability during learning
Robust to uncertainties
Restrictions
No guarantee of policy convergence
Slow policy convergence
Assumes a deterministic policy
Not compatible with continuous action spaces
Tendency to overestimate state-action values
DDQN
Advantages
High stability during learning
Robust to uncertainties
Accurate estimation of state-action values
Restrictions
No guarantee of policy convergence
Slow policy convergence
Assumes a deterministic policy
Not compatible with continuous action spaces
DDPG
Advantages
Rapid convergence towards policy optimum
Assured convergence in complex state-action spaces
Supports stochastic policy learning
Supports continuous action spaces
Restrictions
Instability
Significant sensitivity to noise for parameter learning

Figure 3. Strengths and Shortcomings of Deep Reinforcement Learning Models.

5. Future Directions of Research

Over the past few years, several new and emerging topics have garnered considerable attention and sparked innovative research in the area of deep learning for power engineering. In this section, we discuss several cutting-edge emerging methodologies that can be incorporated into state-of-the-art, data-driven methods for power systems to improve the accuracy of the existing works.

5.1. Attention Mechanism

Attention models have emerged as a groundbreaking topic within the realm of deep learning, revolutionizing the way models process and focus on relevant information. Inspired by the human cognitive process, attention mechanisms enable deep learning models to selectively attend to different parts of input data, assigning varying degrees of importance to each element. This attention-based approach allows models to efficiently capture and leverage crucial features, leading to improved performance across a wide range of tasks. At the heart of attention models lies the attention mechanism, which computes attention weights for different elements of the input based on their relevance to the task at hand. By assigning larger weights to important elements and lower weights to less relevant elements, attention models can dynamically adapt their focus and processing on a per-element basis. This dynamic allocation of attention enables models to effectively handle complex relationships, dependencies, and long-range interactions within the data, thus, enhancing their ability to understand and generate meaningful representations. Moreover, attention models have been widely adopted in various deep learning architectures, including transformer models [226], which have achieved remarkable success in natural language processing tasks. Transformer models leverage self-attention mechanisms to attend to different positions in the input sequence, capturing contextual information and facilitating parallel processing. Beyond text, attention models have also found applications in computer vision [227], speech recognition [228], and reinforcement learning [229], among other areas. Furthermore, the introduction of attention models has significantly advanced the field of deep learning, enabling models to focus on relevant information and achieve state-of-the-art performance on numerous challenging tasks. The ongoing research into attention mechanisms continues to develop novel architectures and techniques that expand the potential of these models and drive innovation in the field of deep learning.

5.2. Transfer Learning and Domain Adaptation

Domain adaptation has emerged as a new topic in deep learning, addressing the challenges posed by the domain shift problem. The domain shift problem arises when a model trained on a specific set of data fails to generalize well to data from a different, but related, domain. In such cases, domain adaptation models aim to leverage the knowledge acquired from the source domain to adapt to the target domain, thereby mitigating the effects of domain shift. The core idea behind domain adaptation models is to learn a shared representation space between the source and target domains in which the domain-specific differences are minimized and the shared characteristics are maximized. This is achieved by optimizing a loss function that simultaneously maximizes the task performance on the source domain while minimizing the discrepancy between the source and target domains. Various techniques have been developed to address the domain adaptation problem, including adversarial training [133], where a domain discriminator is trained to distinguish between the source and target domains, while the model is trained to fool the discriminator by producing domain-invariant features. Another approach is based on discrepancy minimization [230], in which the model is trained to minimize the discrepancy between the distributions of the source and target domains in the feature space. Furthermore, recent research has explored the use of meta-learning and self-supervised learning techniques [231] for domain adaptation, where the model learns to adapt to new domains by leveraging knowledge acquired from previous adaptation tasks or by learning to predict transformations between the domains. Domain adaptation has been applied to various domains, including computer vision [232], natural language processing [233], and speech recognition [234], among others. The ongoing development of domain adaptation models continues to produce novel techniques and architectures that advance the field of deep learning by producing models that better generalize across diverse domains and which achieve improved performance on challenging tasks.

5.3. Interpretable Feature Learning

Interpretable models have emerged as a significant new topic in deep learning, addressing the inherent black-box nature of complex neural networks. As deep learning models become increasingly sophisticated, there is a growing need to understand the decision-making processes and inner workings of these models. Interpretable models aim to bridge this gap by providing explanations and insights into the factors influencing the model's predictions. One approach to interpretability is through feature attribution [235], which assigns importance or relevance scores to input features that indicate their contribution to the model's output. Techniques such as gradient-based methods [236], saliency maps [237], and attention mechanisms [229] help highlight the regions or features that the model focuses on during its decision-making process. Another approach is to learn disentangled representations [238], where the model learns to separate factors of variation in the data, making it easier to understand the underlying relationships. In addition, rule-based models and decision trees [239] provide interpretability by explicitly capturing decision rules and conditions. These models can be trained to mimic the behavior of complex deep learning models, providing a more interpretable alternative. Furthermore, post-hoc interpretability methods aim to interpret black-box models by analyzing their behavior after they have been trained. Techniques such as LIME (Local Interpretable Model-agnostic Explanations) [240] and SHAP (Shapley Additive Explanations) [241] provide explanations at the instance level, highlighting the features that contribute the most to individual predictions. Interpretable models have significant implications, including ethical considerations [242], transparency [243], and trust in AI systems, especially in high-stakes applications such as healthcare [244] and finance [245]. The ongoing advancements in interpretable feature learning continues to produce more reliable and higher-performing models that not only provide accurate inferences but also understandable and explainable insights into how they made those inferences, thus fostering greater transparency into the decision-making process for various domains.

5.4. Physics-Guided Machine Learning

Physics-guided models have recently emerged as a compelling new topic in deep learning, combining the power of data-driven approaches with the foundational principles of physics. These models aim to incorporate prior knowledge and physical laws into the learning process, enabling the development of more accurate and reliable models, particularly in scenarios with limited or noisy data. By integrating physics-based constraints, such as conservation laws, symmetries, or known relationships, deep learning models can capture the underlying structure of the data and make predictions that align with the fundamental principles of the domain [246]. Physics-guided models offer several advantages, including improved generalization, better extrapolation capabilities, and the ability to handle data scarcity. They can also provide interpretability by explicitly incorporating known physics principles [247] into the model architecture, allowing for insights into the decision-making process. Various approaches have been developed to incorporate physics into deep learning models, including physics-informed neural networks [248], where the model is trained to satisfy the physical equations and constraints during the learning process. Another approach involves coupling traditional physics-based models with deep learning architectures, leveraging the strengths of both approaches to achieve more accurate and efficient predictions. Physics-guided models find applications in a wide range of fields, including fluid dynamics [249], materials science [250], medical imaging [251], and climate modeling [252], among others. The ongoing research and development in this area focus on developing new techniques for effectively integrating physics knowledge into deep learning frameworks, enabling the development of robust and interpretable models that align with the laws of nature. By combining the predictive power of deep learning with the foundational understanding of physics, physics-guided models pave the way for advancements in scientific discovery, engineering, and decision-making processes across various domains.

6. Conclusions

With consistent increases in time and storage complexity of problems associated with power system applications, the demand for sophisticated statistical pattern identification techniques has led to the implementation of deep learning-based approaches. This recently developed category of techniques can be classified primarily as discriminative, generative, and reinforcement learning techniques. This paper analyzes the deep discriminative algorithms which offer a precise approach for mapping the complicated input of power system problems to accurate solutions of the supervised problem. Due to their excellent ability for generalization, these models are extensively used for stability evaluation, fault detection, as well as wind and solar power generation forecasting. Then, deep generative methods that offer a probabilistic estimate of data probability densities are discussed. These models can learn complicated probabilistic patterns for a broad range of electrical engineering purposes, such as state estimation, renewable scenario generation, and power grid synthesis. The article concludes with a discussion of deep reinforcement learning algorithms that attempt to optimize an objective using observed rewards captured from the problem's environment. The empirical and mathematical analysis of the adopted methods inspires future studies in the field of deep learning to expand the potential uses of this strong type of framework in new areas of power engineering.

Author Contributions: Conceptualization, M.K. and J.R.; methodology: M.K.; software: M.K.; validation: M.K.; formal analysis: M.K. and J.R.; investigation: M.K. and J.R.; resources: M.K.; data curation: M.K.; writing—original draft preparation: M.K. and J.R.; writing—review and editing: M.K. and J.R.; visualization: J.R.; supervision: M.K.; project administration: M.K. All authors have read and agreed to the published version of the manuscript.

Funding: The research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Lakshmanna, K.; Kaluri, R.; Gundluru, N.; Alzamil, Z.S.; Rajput, D.S.; Khan, A.A.; Haq, M.A.; Alhussen, A. A review on deep learning techniques for IoT data. *Electronics* **2022**, *11*, 1604. [CrossRef]
2. Veeramsetty, V.; Chandra, D.R.; Grimaccia, F.; Mussetta, M. Short term electric power load forecasting using principal component analysis and recurrent neural networks. *Forecasting* **2022**, *4*, 149–164. [CrossRef]
3. Singh, G.; Pal, Y.; Dahiya, A.K. Classification of Power Quality Disturbances using Linear Discriminant Analysis. *Appl. Soft Comput.* **2023**, *138*, 110181. [CrossRef]
4. Hong, J.; Kim, Y.H.; Nhung-Nguyen, H.; Kwon, J.; Lee, H. Deep-Learning Based Fault Events Analysis in Power Systems. *Energies* **2022**, *15*, 5539. [CrossRef]
5. Dumas, J.; Wehenkel, A.; Lanaspeze, D.; Cornélusse, B.; Sutera, A. A deep generative model for probabilistic energy forecasting in power systems: Normalizing flows. *Appl. Energy* **2022**, *305*, 117871. [CrossRef]
6. Wang, S.; Sun, Y.; Zhai, S.; Hou, D.; Wang, P.; Wu, X. Ultra-short-term wind power forecasting based on deep belief network. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 7479–7483.
7. Zhang, Y.; Le, J.; Liao, X.; Zheng, F.; Li, Y. A novel combination forecasting model for wind power integrating least square support vector machine, deep belief network, singular spectrum analysis and locality-sensitive hashing. *Energy* **2019**, *168*, 558–572. [CrossRef]
8. Dairi, A.; Harrou, F.; Sun, Y.; Khadraoui, S. Short-term forecasting of photovoltaic solar power production using variational auto-encoder driven deep learning approach. *Appl. Sci.* **2020**, *10*, 8400. [CrossRef]
9. Massaoudi, M.; Abu-Rub, H.; Refaat, S.S.; Trabelsi, M.; Chihi, I.; Oueslati, F.S. Enhanced deep belief network based on ensemble learning and tree-structured of parzen estimators: An Optimal Photovoltaic Power Forecasting Method. *IEEE Access* **2021**, *9*, 150330–150344. [CrossRef]
10. Wang, C.H.; Lin, K.P.; Lu, Y.M.; Wu, C.F. Deep belief network with seasonal decomposition for solar power output forecasting. *Int. J. Reliab. Qual. Saf. Eng.* **2019**, *26*, 1950029. [CrossRef]
11. Huang, Y.; Xu, Q.; Hu, C.; Sun, Y.; Lin, G. Probabilistic state estimation approach for AC/MTDC distribution system using deep belief network with non-Gaussian uncertainties. *IEEE Sens. J.* **2019**, *19*, 9422–9430. [CrossRef]

12. He, Y.; Chai, S.; Xu, Z.; Lai, C.S.; Xu, X. Power system state estimation using conditional generative adversarial network. *IET Gener. Transm. Distrib.* **2020**, *14*, 5823–5833. [CrossRef]
13. Han, P.; Ellefsen, A.L.; Li, G.; Holmeset, F.T.; Zhang, H. Fault detection with LSTM-based variational autoencoder for maritime components. *IEEE Sens. J.* **2021**, *21*, 21903–21912. [CrossRef]
14. Dehghani, M.; Niknam, T.; Ghiasi, M.; Bayati, N.; Savaghebi, M. Cyber-attack detection in dc microgrids based on deep machine learning and wavelet singular values approach. *Electronics* **2021**, *10*, 1914. [CrossRef]
15. Chen, H.; Wang, Y.H.; Fan, C.H. A convolutional autoencoder-based approach with batch normalization for energy disaggregation. *J. Supercomput.* **2021**, *77*, 2961–2978. [CrossRef]
16. Aravind, P.; Sarath, T. Non-Intrusive Load Monitoring for Energy Consumption Disaggregation. In Proceedings of the 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 20–22 October 2022; pp. 14–19.
17. Tsakoumis, A.; Vladov, S.; Mladenov, V. Electric load forecasting with multilayer perceptron and Elman neural network. In Proceedings of the 6th Seminar on Neural Network Applications in Electrical Engineering, Belgrade, Yugoslavia, 26–28 September 2002; pp. 87–90. [CrossRef]
18. Wan, C.; Xu, Z.; Pinson, P.; Dong, Z.Y.; Wong, K.P. Probabilistic forecasting of wind power generation using extreme learning machine. *IEEE Trans. Power Syst.* **2013**, *29*, 1033–1044. [CrossRef]
19. Yang, H.; Yi, J.; Zhao, J.; Dong, Z. Extreme learning machine based genetic algorithm and its application in power system economic dispatch. *Neurocomputing* **2013**, *102*, 154–162. [CrossRef]
20. Villa-Acevedo, W.M.; López-Lezama, J.M.; Colomé, D.G.; Cepeda, J. Long-term voltage stability monitoring of power system areas using a kernel extreme learning machine approach. *Alex. Eng. J.* **2022**, *61*, 1353–1367. [CrossRef]
21. Sheikhlar, Z.; Hedayati, M.; Tafti, A.D.; Farahani, H.F. Fuzzy Elman Wavelet Network: Applications to function approximation, system identification, and power system control. *Inf. Sci.* **2022**, *583*, 306–331. [CrossRef]
22. Yousef, H.; Soliman, H.M.; Albadi, M. Nonlinear power system excitation control using adaptive wavelet networks. *Neurocomputing* **2017**, *230*, 302–311. [CrossRef]
23. Czumbil, L.; Micu, D.D.; Stet, D.; Ceclan, A. A neural network approach for the inductive coupling between overhead power lines and nearby metallic pipelines. In Proceedings of the 2016 International Symposium on Fundamentals of Electrical Engineering (ISFEE), Bucharest, Romania, 30 June–2 July 2016; pp. 1–6. [CrossRef]
24. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef] [PubMed]
25. Su, T.; Liu, Y.; Zhao, J.; Liu, J. Deep belief network enabled surrogate modeling for fast preventive control of power system transient stability. *IEEE Trans. Ind. Inform.* **2021**, *18*, 315–326. [CrossRef]
26. Cui, M.; Khodayar, M.; Chen, C.; Wang, X.; Zhang, Y.; Khodayar, M.E. Deep learning-based time-varying parameter identification for system-wide load modeling. *IEEE Trans. Smart Grid* **2019**, *10*, 6102–6114. [CrossRef]
27. Hua, Y.; Guo, J.; Zhao, H. Deep belief networks and deep learning. In Proceedings of the 2015 International Conference on Intelligent Computing and Internet of Things, Harbin, China, 17–18 January 2015; pp. 1–4.
28. Zhong, Z.; Guan, L.; Su, Y.; Yu, J.; Huang, J.; Guo, M. A method of multivariate short-term voltage stability assessment based on heterogeneous graph attention deep network. *Int. J. Electr. Power Energy Syst.* **2022**, *136*, 107648. [CrossRef]
29. Cai, H.; Hill, D.J. A data-driven distributed and easy-to-transfer method for short-term voltage stability assessment. *Int. J. Electr. Power Energy Syst.* **2022**, *139*, 107960. [CrossRef]
30. Dorado-Rojas, S.A.; Bogodorova, T.; Vanfretti, L. Time Series-Based Small-Signal stability assessment using deep learning. In Proceedings of the 2021 North American Power Symposium (NAPS), College Station, TX, USA, 14–16 November 2021; pp. 1–6.
31. Haque, M.; Shaheed, M.N.; Choi, S. Deep learning based micro-grid fault detection and classification in future smart vehicle. In Proceedings of the 2018 IEEE Transportation Electrification Conference and Expo (ITEC), Long Beach, CA, USA, 13–15 June 2018; pp. 1082–1087.
32. Pavlovski, M.; Alqudah, M.; Dokic, T.; Hai, A.A.; Kezunovic, M.; Obradovic, Z. Hierarchical convolutional neural networks for event classification on PMU measurements. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]
33. Rick, R.; Berton, L. Energy forecasting model based on CNN-LSTM-AE for many time series with unequal lengths. *Eng. Appl. Artif. Intell.* **2022**, *113*, 104998. [CrossRef]
34. Gensler, A.; Henze, J.; Sick, B.; Raabe, N. Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 002858–002865.
35. Wang, Y.; Yang, H.; Yuan, X.; Shardt, Y.A.; Yang, C.; Gui, W. Deep learning for fault-relevant feature extraction and fault classification with stacked supervised auto-encoder. *J. Process. Control.* **2020**, *92*, 79–89. [CrossRef]
36. Sarajcev, P.; Kunac, A.; Petrovic, G.; Despalatovic, M. Power system transient stability assessment using stacked autoencoder and voting ensemble. *Energies* **2021**, *14*, 3148. [CrossRef]
37. Pepiciello, A.; Vaccaro, A. Artificial Neural Network-based Small Signal Stability Analysis of Power Systems. In Proceedings of the 2021 IEEE Madrid PowerTech, Madrid, Spain, 28 June–2 July 2021; pp. 1–5. [CrossRef]
38. Yang, F.; Liu, J.; Lu, B.; Chai, W.; Ren, L. Power Supply Reliability Evaluation Method of Distribution Network Based on Improved LSTM Neural Network. In Proceedings of the 2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 9–11 December 2022; pp. 460–463.

39. ul Islam, B.; Ahmed, S.F. Short-term electrical load demand forecasting based on lstm and rnn deep neural networks. *Math. Probl. Eng.* **2022**, *2022*, 2316474. [CrossRef]
40. Niu, D.; Yu, M.; Sun, L.; Gao, T.; Wang, K. Short-term multi-energy load forecasting for integrated energy systems based on CNN-BiGRU optimized by attention mechanism. *Appl. Energy* **2022**, *313*, 118801. [CrossRef]
41. Qing, X.; Niu, Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy* **2018**, *148*, 461–468. [CrossRef]
42. Xiang, L.; Wang, P.; Yang, X.; Hu, A.; Su, H. Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism. *Measurement* **2021**, *175*, 109094. [CrossRef]
43. Wen, S.; Wang, Y.; Tang, Y.; Xu, Y.; Li, P.; Zhao, T. Real-time identification of power fluctuations based on LSTM recurrent neural network: A case study on Singapore power system. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5266–5275. [CrossRef]
44. Cui, C.; He, M.; Di, F.; Lu, Y.; Dai, Y.; Lv, F. Research on power load forecasting method based on LSTM model. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 1657–1660.
45. Khodayar, M.; Wang, J.; Wang, Z. Energy disaggregation via deep temporal dictionary learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 1696–1709. [CrossRef]
46. Jalali, S.M.J.; Ahmadian, S.; Khodayar, M.; Khosravi, A.; Ghasemi, V.; Shafie-khah, M.; Nahavandi, S.; Catalão, J.P. Towards novel deep neuroevolution models: Chaotic levy grasshopper optimization for short-term wind speed forecasting. *Eng. Comput.* **2021**, *38*, 1787–1811. [CrossRef]
47. Belagoune, S.; Bali, N.; Bakdi, A.; Baadji, B.; Atif, K. Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems. *Measurement* **2021**, *177*, 109330. [CrossRef]
48. Yin, L.; Xie, J. Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems. *Appl. Energy* **2021**, *283*, 116328. [CrossRef]
49. Jalali, S.M.J.; Ahmadian, S.; Khodayar, M.; Khosravi, A.; Shafie-khah, M.; Nahavandi, S.; Catalao, J.P. An advanced short-term wind power forecasting framework based on the optimized deep neural network models. *Int. J. Electr. Power Energy Syst.* **2022**, *141*, 108143. [CrossRef]
50. Jalali, S.M.J.; Khodayar, M.; Ahmadian, S.; Noman, M.K.; Khosravi, A.; Islam, S.M.S.; Wang, F.; Catalão, J.P. A new uncertainty-aware deep neuroevolution model for quantifying tidal prediction. In Proceedings of the 2021 IEEE Industry Applications Society Annual Meeting (IAS), Vancouver, BC, Canada, 10–14 October 2021; pp. 1–6.
51. Hou, J.; Xie, C.; Wang, T.; Yu, Z.; Lü, Y.; Dai, H. Power system transient stability assessment based on voltage phasor and convolution neural network. In Proceedings of the 2018 IEEE International Conference on Energy Internet (ICEI), Beijing, China, 21–25 May 2018; pp. 247–251.
52. Zhi, Z.; Liu, L.; Liu, D.; Hu, C. Fault detection of the harmonic reducer based on CNN-LSTM with a novel denoising algorithm. *IEEE Sens. J.* **2021**, *22*, 2572–2581. [CrossRef]
53. He, C.; Ge, D.; Yang, M.; Yong, N.; Wang, J.; Yu, J. A data-driven adaptive fault diagnosis methodology for nuclear power systems based on NSGAII-CNN. *Ann. Nucl. Energy* **2021**, *159*, 108326. [CrossRef]
54. Rizvi, S.M.H.; Sadanandan, S.K.; Srivastava, A.K. Data-driven short-term voltage stability assessment using convolutional neural networks considering data anomalies and localization. *IEEE Access* **2021**, *9*, 128345–128358. [CrossRef]
55. Tian, J.; Sun, X.; Du, Y.; Zhao, S.; Liu, Q.; Zhang, K.; Yi, W.; Huang, W.; Wang, C.; Wu, X.; et al. Recent advances for quantum neural networks in generative learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *early access*. [CrossRef]
56. Khamparia, A.; Singh, K.M. A systematic review on deep learning architectures and applications. *Expert Syst.* **2019**, *36*, e12400. [CrossRef]
57. Khodayar, M.; Wang, J.; Manthouri, M. Interval deep generative neural network for wind speed forecasting. *IEEE Trans. Smart Grid* **2018**, *10*, 3974–3989. [CrossRef]
58. Zheng, L.; Hu, W.; Zhou, Y.; Min, Y.; Xu, X.; Wang, C.; Yu, R. Deep belief network based nonlinear representation learning for transient stability assessment. In Proceedings of the 2017 IEEE Power & Energy Society General Meeting, Chicago, IL, USA, 16–20 July 2017; pp. 1–5.
59. Dedinec, A.; Filiposka, S.; Dedinec, A.; Kocarev, L. Deep belief network based electricity load forecasting: An analysis of Macedonian case. *Energy* **2016**, *115*, 1688–1700. [CrossRef]
60. Mestav, K.R.; Luengo-Rozas, J.; Tong, L. Bayesian state estimation for unobservable distribution systems via deep learning. *IEEE Trans. Power Syst.* **2019**, *34*, 4910–4920. [CrossRef]
61. Su, Y.; Meng, L.; Kong, X.; Xu, T.; Lan, X.; Li, Y. Small sample fault diagnosis method for wind turbine gearbox based on optimized generative adversarial networks. *Eng. Fail. Anal.* **2022**, *140*, 106573. [CrossRef]
62. Adiban, M.; Safari, A.; Salvi, G. Step-gan: A step-by-step training for multi generator gans with application to cyber security in power systems. *arXiv* **2020**, arXiv:2009.05184.
63. Jiang, C.; Mao, Y.; Chai, Y.; Yu, M.; Tao, S. Scenario generation for wind power using improved generative adversarial networks. *IEEE Access* **2018**, *6*, 62193–62203. [CrossRef]
64. Khazeiynasab, S.R.; Zhao, J.; Batarseh, I.; Tan, B. Power plant model parameter calibration using conditional variational autoencoder. *IEEE Trans. Power Syst.* **2021**, *37*, 1642–1652. [CrossRef]

65. Pan, Z.; Wang, J.; Liao, W.; Chen, H.; Yuan, D.; Zhu, W.; Fang, X.; Zhu, Z. Data-driven EV load profiles generation using a variational auto-encoder. *Energies* **2019**, *12*, 849. [CrossRef]
66. Duan, J.; Shi, D.; Diao, R.; Li, H.; Wang, Z.; Zhang, B.; Bian, D.; Yi, Z. Deep-reinforcement-learning-based autonomous voltage control for power grid operations. *IEEE Trans. Power Syst.* **2019**, *35*, 814–817. [CrossRef]
67. Huang, Q.; Huang, R.; Hao, W.; Tan, J.; Fan, R.; Huang, Z. Adaptive power system emergency control using deep reinforcement learning. *IEEE Trans. Smart Grid* **2019**, *11*, 1171–1182. [CrossRef]
68. Tang, X.; Chen, J.; Pu, H.; Liu, T.; Khajepour, A. Double deep reinforcement learning-based energy management for a parallel hybrid electric vehicle with engine start–stop strategy. *IEEE Trans. Transp. Electrif.* **2021**, *8*, 1376–1388. [CrossRef]
69. Ye, Y.; Qiu, D.; Sun, M.; Papadaskalopoulos, D.; Strbac, G. Deep reinforcement learning for strategic bidding in electricity markets. *IEEE Trans. Smart Grid* **2019**, *11*, 1343–1355. [CrossRef]
70. Wang, B.; Li, Y.; Ming, W.; Wang, S. Deep reinforcement learning method for demand response management of interruptible load. *IEEE Trans. Smart Grid* **2020**, *11*, 3146–3155. [CrossRef]
71. Jurj, D.I.; Czumbil, L.; Bârgăuan, B.; Ceclan, A.; Polycarpou, A.; Micu, D.D. Custom Outlier Detection for Electrical Energy Consumption Data Applied in Case of Demand Response in Block of Buildings. *Sensors* **2021**, *21*, 2946. [CrossRef]
72. Crețu, M.; Czumbil, L.; Bârgăuan, B.; Ceclan, A.; Berciu, A.; Polycarpou, A.; Rizzo, R.; Micu, D.D. Modelling and evaluation of the Baseline Energy Consumption and the Key Performance Indicators in Technical University of Cluj-Napoca buildings within a Demand Response programme: A case study. *IET Renew. Power Gener.* **2020**, *14*, 2864–2875. [CrossRef]
73. Li, Y.; Wang, R.; Li, Y.; Zhang, M.; Long, C. Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach. *Appl. Energy* **2023**, *329*, 120291. [CrossRef]
74. Wei, F.; Wan, Z.; He, H. Cyber-attack recovery strategy for smart grid based on deep reinforcement learning. *IEEE Trans. Smart Grid* **2019**, *11*, 2476–2486. [CrossRef]
75. Alqahtani, M.; Hu, M. Dynamic energy scheduling and routing of multiple electric vehicles using deep reinforcement learning. *Energy* **2022**, *244*, 122626. [CrossRef]
76. Khodayar, M.; Kaynak, O.; Khodayar, M.E. Rough deep neural architecture for short-term wind speed forecasting. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2770–2779. [CrossRef]
77. Khodayar, M.; Wang, J. Spatio-temporal graph deep neural network for short-term wind speed forecasting. *IEEE Trans. Sustain. Energy* **2018**, *10*, 670–681. [CrossRef]
78. Bailly, A.; Blanc, C.; Francis, É.; Guillotin, T.; Jamal, F.; Wakim, B.; Roy, P. Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Comput. Methods Programs Biomed.* **2022**, *213*, 106504. [CrossRef]
79. Khodayar, M.; Mohammadi, S.; Khodayar, M.E.; Wang, J.; Liu, G. Convolutional graph autoencoder: A generative deep neural network for probabilistic spatio-temporal solar irradiance forecasting. *IEEE Trans. Sustain. Energy* **2019**, *11*, 571–583. [CrossRef]
80. Desai, M.; Shah, M. An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). *Clin. eHealth* **2021**, *4*, 1–11. [CrossRef]
81. Khodayar, M.; Liu, G.; Wang, J.; Khodayar, M.E. Deep learning in power systems research: A review. *CSEE J. Power Energy Syst.* **2020**, *7*, 209–220.
82. Kumar, S.K. On weight initialization in deep neural networks. *arXiv* **2017**, arXiv:1704.08863.
83. Khodayar, M.; Teshnehlab, M. Robust deep neural network for wind speed prediction. In Proceedings of the 2015 4th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), Zahedan, Iran, 9–11 September 2015; pp. 1–5.
84. Hanin, B.; Rolnick, D. Deep relu networks have surprisingly few activation patterns. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
85. Douglas, S.C.; Yu, J. Why RELU units sometimes die: Analysis of single-unit error backpropagation in neural networks. In Proceedings of the 2018 52nd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 28–31 October 2018; pp. 864–868.
86. Xu, J.; Li, Z.; Du, B.; Zhang, M.; Liu, J. Reluplex made more practical: Leaky ReLU. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–7.
87. Dong, Z.; Hou, K.; Meng, H.; Yu, X.; Jia, H. Data-driven power system reliability evaluation based on stacked denoising auto-encoders. *Energy Rep.* **2022**, *8*, 920–927. [CrossRef]
88. Jiang, Y.; Liu, M.; Peng, H.; Bhuiyan, M.Z.A. A reliable deep learning-based algorithm design for IoT load identification in smart grid. *Ad Hoc Netw.* **2021**, *123*, 102643. [CrossRef]
89. Xiong, B.; Lou, L.; Meng, X.; Wang, X.; Ma, H.; Wang, Z. Short-term wind power forecasting based on Attention Mechanism and Deep Learning. *Electr. Power Syst. Res.* **2022**, *206*, 107776. [CrossRef]
90. Wang, K.; Qi, X.; Liu, H. Photovoltaic power forecasting based LSTM-Convolutional Network. *Energy* **2019**, *189*, 116225. [CrossRef]
91. Shadi, M.R.; Ameli, M.T.; Azad, S. A real-time hierarchical framework for fault detection, classification, and location in power systems using PMUs data and deep learning. *Int. J. Electr. Power Energy Syst.* **2022**, *134*, 107399. [CrossRef]
92. Chen, Y.; Xie, H.; Lu, G.; Zhang, L.; Yan, J.; Li, S. Transient voltage stability assessment of renewable energy grid based on residual SDE-Net. *Energy Rep.* **2022**, *8*, 991–1001. [CrossRef]
93. Tapia, E.A.; Colomé, D.G.; Rueda Torres, J.L. Recurrent Convolutional Neural Network-Based Assessment of Power System Transient Stability and Short-Term Voltage Stability. *Energies* **2022**, *15*, 9240. [CrossRef]

94. Dai, L.; Guo, J.; Wan, J.L.; Wang, J.; Zan, X. A reliability evaluation model of rolling bearings based on WKN-BiGRU and Wiener process. *Reliab. Eng. Syst. Saf.* **2022**, *225*, 108646. [CrossRef]
95. Kamruzzaman, M.; Bhusal, N.; Benidris, M. A convolutional neural network-based approach to composite power system reliability evaluation. *Int. J. Electr. Power Energy Syst.* **2022**, *135*, 107468. [CrossRef]
96. Wang, Y.; Lu, C.; Zhang, X. Convolution neural network based load model parameter selection considering short-term voltage stability. *CSEE J. Power Energy Syst.* **2022**, *early access*.
97. Abdel-Basset, M.; Hawash, H.; Sallam, K.; Askar, S.; Abouhawwash, M. STLF-Net: Two-stream deep network for short-term load forecasting in residential buildings. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 4296–4311. [CrossRef]
98. Butt, F.; Hussain, L.; Jafri, S.; Lone, K.; Alajmi, M.; Abunadi, I.; Al-Wesabi, F.; Hamza, M. Optimizing parameters of artificial intelligence deep convolutional neural networks (CNN) to improve prediction performance of load forecasting system. In *IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Riyadh, Saudi Arabia, 2022; Volume 1026, p. 012028.
99. Chung, W.H.; Gu, Y.H.; Yoo, S.J. District heater load forecasting based on machine learning and parallel CNN-LSTM attention. *Energy* **2022**, *246*, 123350. [CrossRef]
100. Ijaz, K.; Hussain, Z.; Ahmad, J.; Ali, S.F.; Adnan, M.; Khosa, I. A novel temporal feature selection based lstm model for electrical short-term load forecasting. *IEEE Access* **2022**, *10*, 82596–82613. [CrossRef]
101. Javed, U.; Ijaz, K.; Jawad, M.; Khosa, I.; Ansari, E.A.; Zaidi, K.S.; Rafiq, M.N.; Shabbir, N. A novel short receptive field based dilated causal convolutional network integrated with Bidirectional LSTM for short-term load forecasting. *Expert Syst. Appl.* **2022**, *205*, 117689. [CrossRef]
102. Darab, C.; Antoniu, T.; Beleiu, H.G.; Pavel, S.; Birou, I.; Micu, D.D.; Ungureanu, S.; Cirstea, S.D. Hybrid Load Forecasting Using Gaussian Process Regression and Novel Residual Prediction. *Appl. Sci.* **2020**, *10*, 4588. [CrossRef]
103. Shafiei Chafi, Z.; Afrakhte, H. Short-term load forecasting using neural network and particle swarm optimization (PSO) algorithm. *Math. Probl. Eng.* **2021**, *2021*, 1–10. [CrossRef]
104. Zhang, Y.; Zhou, S.; Zhang, Z.; Yan, L.; Liu, L. Design of an ultra-short-term wind power forecasting model combined with CNN and LSTM networks. In Proceedings of the Recent Developments in Intelligent Computing, Communication and Devices: Proceedings of ICCD 2019 5, Xi'an, China, 22–24 November 2021; pp. 141–145.
105. Garg, S.; Krishnamurthi, R. A CNN Encoder Decoder LSTM Model for Sustainable Wind Power Predictive Analytics. *Sustain. Comput. Inform. Syst.* **2023**, *38*, 100869. [CrossRef]
106. Agga, A.; Abbou, A.; Labbadi, M.; El Houm, Y.; Ali, I.H.O. CNN-LSTM: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production. *Electr. Power Syst. Res.* **2022**, *208*, 107908. [CrossRef]
107. Jalali, S.M.J.; Ahmadian, S.; Kavousi-Fard, A.; Khosravi, A.; Nahavandi, S. Automated deep CNN-LSTM architecture design for solar irradiance forecasting. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 54–65. [CrossRef]
108. Verma, S.; Singh, S.; Majumdar, A. Multi-label LSTM autoencoder for non-intrusive appliance load monitoring. *Electr. Power Syst. Res.* **2021**, *199*, 107414. [CrossRef]
109. Angelis, G.F.; Timplalexis, C.; Salamanis, A.I.; Krinidis, S.; Ioannidis, D.; Kehagias, D.; Tzovaras, D. Energformer: A New Transformer Model for Energy Disaggregation. *IEEE Trans. Consum. Electron.* **2023**, *early access*. [CrossRef]
110. Dash, S.; Sahoo, N. Deep Sequence-to-point Learning for Electric Appliance Energy Disaggregation in Smart Building. In Proceedings of the 2022 IEEE Region 10 Symposium (TENSYMP), Mumbai, India, 1–3 July 2022; pp. 1–6.
111. MansourLakouraj, M.; Gautam, M.; Livani, H.; Benidris, M. A multi-rate sampling PMU-based event classification in active distribution grids with spectral graph neural network. *Electr. Power Syst. Res.* **2022**, *211*, 108145. [CrossRef]
112. Yuan, Y.; Wang, Z.; Wang, Y. Learning Latent Interactions for Event Classification via Graph Neural Networks and PMU Data. *IEEE Trans. Power Syst.* **2022**, *38*, 617–629. [CrossRef]
113. Ma, H.; Lei, X.; Li, Z.; Yu, S.; Liu, B.; Dong, X. Deep-learning based Power System Events Detection Technology Using Spatio-temporal and Frequency Information. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2023**, *early access*. [CrossRef]
114. Ahmed, A.; Sadanandan, S.K.; Pandey, S.; Basumallik, S.; Srivastava, A.K.; Wu, Y. Event Analysis in Transmission Systems Using Spatial Temporal Graph Encoder Decoder (STGED). *IEEE Trans. Power Syst.* **2022**, *early access*. [CrossRef]
115. Li, Z.; He, Y.; Xing, Z.; Duan, J. Transformer fault diagnosis based on improved deep coupled dense convolutional neural network. *Electr. Power Syst. Res.* **2022**, *209*, 107969. [CrossRef]
116. Junior, R.F.R.; dos Santos Areias, I.A.; Campos, M.M.; Teixeira, C.E.; da Silva, L.E.B.; Gomes, G.F. Fault detection and diagnosis in electric motors using 1d convolutional neural networks with multi-channel vibration signals. *Measurement* **2022**, *190*, 110759. [CrossRef]
117. Rahimilarki, R.; Gao, Z.; Jin, N.; Zhang, A. Convolutional neural network fault classification based on time-series analysis for benchmark wind turbine machine. *Renew. Energy* **2022**, *185*, 916–931. [CrossRef]
118. Cordoni, F.; Bacchiega, G.; Bondani, G.; Radu, R.; Muradore, R. A multi–modal unsupervised fault detection system based on power signals and thermal imaging via deep AutoEncoder neural network. *Eng. Appl. Artif. Intell.* **2022**, *110*, 104729. [CrossRef]
119. Khodayar, M.; Mohammadi, S.; Khodayar, M.; Wang, J.; Liu, G. Convolutional graph auto-encoder: A deep generative neural architecture for probabilistic spatio-temporal solar irradiance forecasting. *arXiv* **2018**, arXiv:1809.03538.
120. Saffari, M.; Khodayar, M.; Jalali, S.M.J.; Shafie-khah, M.; Catalão, J.P. Deep convolutional graph rough variational auto-encoder for short-term photovoltaic power forecasting. In Proceedings of the 2021 International Conference on Smart Energy Systems and Technologies (SEST), Vaasa, Finland, 6–8 September 2021; pp. 1–6.

121. Khodayar, M.; Wang, J.; Wang, Z. Deep generative graph distribution learning for synthetic power grids. *arXiv* **2019**, arXiv:1901.09674.
122. Saffari, M.; Khodayar, M.; Khodayar, M.E. Deep recurrent extreme learning machine for behind-the-meter photovoltaic disaggregation. *Electr. J.* **2022**, *35*, 107137. [CrossRef]
123. Saffari, M.; Williams, M.; Khodayar, M.; Shafie-khah, M.; Catalão, J.P. Robust wind speed forecasting: A deep spatio-temporal approach. In Proceedings of the 2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Bari, Italy, 7–10 September 2021; pp. 1–6.
124. Ahmed, H.; Ullah, A. Exponential moving average extended kalman filter for robust battery state-of-charge estimation. In Proceedings of the 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), IEEE, Chittagong, Bangladesh, 26–27 February 2022; pp. 555–560.
125. Karim, S.A.; Alwi, S.A. Electricity load forecasting in UTP using moving averages and exponential smoothing techniques. *Appl. Math. Sci.* **2013**, *7*, 4003–4014. [CrossRef]
126. Wang, S.; Gao, W.; Meliopoulos, A.S. An alternative method for power system dynamic state estimation based on unscented transform. *IEEE Trans. Power Syst.* **2011**, *27*, 942–950. [CrossRef]
127. Li, Z. Hilbert-huang transform based application in power system fault detection. In Proceedings of the 2009 International Workshop on Intelligent Systems and Applications, Wuhan, China, 23–24 May 2009; pp. 1–4.
128. Liu, Y.; Sun, K.; Yao, R.; Wang, B. Power system time domain simulation using a differential transformation method. *IEEE Trans. Power Syst.* **2019**, *34*, 3739–3748. [CrossRef]
129. Yu, Z.; Niu, Z.; Tang, W.; Wu, Q. Deep learning for daily peak load forecasting–a novel gated recurrent neural network combining dynamic time warping. *IEEE Access* **2019**, *7*, 17184–17194. [CrossRef]
130. Yang, R.; Zhang, D.; Li, Z.; Yang, K.; Mo, S.; Li, L. Mechanical fault diagnostics of power transformer on-load tap changers using dynamic time warping. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 3119–3127. [CrossRef]
131. Banna, H.U.; Yu, Z.; Shi, D.; Wang, Z.; Su, D.; Xu, C.; Solanki, S.K.; Solanki, J.M. Online coherence identification using dynamic time warping for controlled islanding. *J. Mod. Power Syst. Clean Energy* **2019**, *7*, 38–54. [CrossRef]
132. Jalali, S.M.J.; Khodayar, M.; Khosravi, A.; Osório, G.J.; Nahavandi, S.; Catalão, J.P. An advanced generative deep learning framework for probabilistic spatio-temporal wind power forecasting. In Proceedings of the 2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Bari, Italy, 7–10 September 2021; pp. 1–6.
133. Saffari, M.; Khodayar, M.; Jalali, S.M.J. Sparse Adversarial Unsupervised Domain Adaptation With Deep Dictionary Learning for Traffic Scene Classification. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *early access*. [CrossRef]
134. Khodayar, M.; Wang, J. Probabilistic time-varying parameter identification for load modeling: A deep generative approach. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1625–1636. [CrossRef]
135. Khodayar, M.; Saffari, M.; Williams, M.; Jalali, S.M.J. Interval deep learning architecture with rough pattern recognition and fuzzy inference for short-term wind speed forecasting. *Energy* **2022**, *254*, 124143. [CrossRef]
136. Khodayar, M.; Wang, J. Deep generative graph learning for power grid synthesis. In Proceedings of the 2021 International Conference on Smart Energy Systems and Technologies (SEST), Vaasa, Finland, 6–8 September 2021; pp. 1–6.
137. Ruiz, F.; Titsias, M. A contrastive divergence for combining variational inference and mcmc. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 5537–5545.
138. Wu, S.; Zheng, L.; Hu, W.; Yu, R.; Liu, B. Improved deep belief network and model interpretation method for power system transient stability assessment. *J. Mod. Power Syst. Clean Energy* **2019**, *8*, 27–37. [CrossRef]
139. Ouyang, T.; He, Y.; Li, H.; Sun, Z.; Baek, S. Modeling and forecasting short-term power load with copula model and deep belief network. *IEEE Trans. Emerg. Top. Comput. Intell.* **2019**, *3*, 127–136. [CrossRef]
140. Wei, Y.; Zhang, H.; Dai, J.; Zhu, R.; Qiu, L.; Dong, Y.; Fang, S. Deep Belief Network with Swarm Spider Optimization Method for Renewable Energy Power Forecasting. *Processes* **2023**, *11*, 1001. [CrossRef]
141. Lu, K.D.; Zeng, G.Q.; Luo, X.; Weng, J.; Luo, W.; Wu, Y. Evolutionary deep belief network for cyber-attack detection in industrial automation and control system. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7618–7627. [CrossRef]
142. Li, B.; Wu, J. Adaptive Assessment of Power System Transient Stability Based on Active Transfer Learning With Deep Belief Network. *IEEE Trans. Autom. Sci. Eng.* **2022**, *early access*. [CrossRef]
143. Yang, H.; Qiu, R.C.; Shi, X.; He, X. Unsupervised feature learning for online voltage stability evaluation and monitoring based on variational autoencoder. *Electr. Power Syst. Res.* **2020**, *182*, 106253. [CrossRef]
144. Li, J.; Yang, H.; Yan, L.; Li, Z.; Liu, D.; Xia, Y. Data augment using deep convolutional generative adversarial networks for transient stability assessment of power systems. In Proceedings of the 2020 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020; pp. 6135–6140.
145. Moradzadeh, A.; Moayyed, H.; Zare, K.; Mohammadi-Ivatloo, B. Short-term electricity demand forecasting via variational autoencoders and batch training-based bidirectional long short-term memory. *Sustain. Energy Technol. Assess.* **2022**, *52*, 102209. [CrossRef]
146. Liang, Y.; Zhi, L.; Haiwei, Y. Medium-Term Load Forecasting Method With Improved Deep Belief Network for Renewable Energy. *Distrib. Gener. Altern. Energy J.* **2022**, *37*, 485–500. [CrossRef]
147. Fan, C.; Ding, C.; Zheng, J.; Xiao, L.; Ai, Z. Empirical mode decomposition based multi-objective deep belief network for short-term power load forecasting. *Neurocomputing* **2020**, *388*, 110–123. [CrossRef]

148. Mukaroh, A.; Le, T.T.H.; Kim, H. Background load denoising across complex load based on generative adversarial network to enhance load identification. *Sensors* **2020**, *20*, 5674. [CrossRef]
149. Tian, C.; Ye, Y.; Lou, Y.; Zuo, W.; Zhang, G.; Li, C. Daily power demand prediction for buildings at a large scale using a hybrid of physics-based model and generative adversarial network. In *Building Simulation*; Tsinghua University Press: Beijing, China, 2022; Volume 15, pp. 1685–1701.
150. Kosana, V.; Teeparthi, K.; Madasthu, S. A novel and hybrid framework based on generative adversarial network and temporal convolutional approach for wind speed prediction. *Sustain. Energy Technol. Assess.* **2022**, *53*, 102467. [CrossRef]
151. Zhou, B.; Duan, H.; Wu, Q.; Wang, H.; Or, S.W.; Chan, K.W.; Meng, Y. Short-term prediction of wind power and its ramp events based on semi-supervised generative adversarial network. *Int. J. Electr. Power Energy Syst.* **2021**, *125*, 106411. [CrossRef]
152. Hu, W.; Zhang, X.; Zhu, L.; Li, Z. Short-term photovoltaic power prediction based on similar days and improved SOA-DBN model. *IEEE Access* **2020**, *9*, 1958–1971. [CrossRef]
153. Lei, L.; Guo, J.; Wang, F.; Zhang, L. Short-Term Prediction of Photovoltaic Power Generation Based on Deep Belief Network with Momentum Factor. In Proceedings of the 2020 Chinese Intelligent Systems Conference: Volume I, Fuzhou, China, 16–17 October 2021; pp. 778–791.
154. Tang, R.; Dore, J.; Ma, J.; Leong, P.H. Interpolating high granularity solar generation and load consumption data using super resolution generative adversarial network. *Appl. Energy* **2021**, *299*, 117297. [CrossRef]
155. Zhang, W.; Luo, Y.; Zhang, Y.; Srinivasan, D. SolarGAN: Multivariate solar data imputation using generative adversarial network. *IEEE Trans. Sustain. Energy* **2020**, *12*, 743–746. [CrossRef]
156. Huang, X.; Li, Q.; Tai, Y.; Chen, Z.; Liu, J.; Shi, J.; Liu, W. Time series forecasting for hourly photovoltaic power using conditional generative adversarial network and Bi-LSTM. *Energy* **2022**, *246*, 123403. [CrossRef]
157. Zhao, L.; Liu, Y.; Zhao, J.; Zhang, Y.; Xu, L.; Xiang, Y.; Liu, J. Robust PCA-deep belief network surrogate model for distribution system topology identification with DERs. *Int. J. Electr. Power Energy Syst.* **2021**, *125*, 106441. [CrossRef]
158. Wang, X.; Cui, P.; Du, Y.; Yang, Y. Variational autoencoder based fault detection and location method for power distribution network. In Proceedings of the 2020 8th International Conference on Condition Monitoring and Diagnosis (CMD), Phuket, Thailand, 25–28 October 2020; pp. 282–285.
159. Biswas, S.; Meyur, R.; Centeno, V.A. Devlearn: A deep visual learning framework for determining the location of temporary faults in power systems. In Proceedings of the 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Tempe, AZ, USA, 11–13 November 2020; pp. 1–6.
160. Hassani, H.; Razavi-Far, R.; Saif, M.; Palade, V. Generative adversarial network-based scheme for diagnosing faults in cyber-physical power systems. *Sensors* **2021**, *21*, 5173. [CrossRef]
161. Lu, F.; Niu, R.; Zhang, Z.; Guo, L.; Chen, J. A generative adversarial network-based fault detection approach for photovoltaic panel. *Appl. Sci.* **2022**, *12*, 1789. [CrossRef]
162. Aligholian, A.; Shahsavari, A.; Cortez, E.; Stewart, E.; Mohsenian-Rad, H. Event detection in micro-pmu data: A generative adversarial network scoring method. In Proceedings of the 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, 2–6 August 2020; pp. 1–5.
163. Li, B.; Cheng, F.; Cai, H.; Zhang, X.; Cai, W. A semi-supervised approach to fault detection and diagnosis for building HVAC systems based on the modified generative adversarial network. *Energy Build.* **2021**, *246*, 111044. [CrossRef]
164. Guo, Q.; Li, Y.; Song, Y.; Wang, D.; Chen, W. Intelligent fault diagnosis method based on full 1-D convolutional generative adversarial network. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2044–2053. [CrossRef]
165. Lee, Y.O.; Jo, J.; Hwang, J. Application of deep neural network and generative adversarial network to industrial maintenance: A case study of induction motor fault detection. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 1–14 December 2017; pp. 3248–3253.
166. Zheng, R.; Gu, J. Anomaly detection for power system forecasting under data corruption based on variational auto-encoder. In Proceedings of the IET Conference Proceedings, Shanghai, China, 24–25 October 2019; pp. 206–212. [CrossRef]
167. Ding, Y.; Ma, K.; Pu, T.; Wang, X.; Li, R.; Zhang, D. A deep learning-based classification scheme for cyber-attack detection in power system. *IET Energy Syst. Integr.* **2021**, *3*, 274–284. [CrossRef]
168. Durairaj, D.; Venkatasamy, T.K.; Mehbodniya, A.; Umar, S.; Alam, T. Intrusion detection and mitigation of attacks in microgrid using enhanced deep belief network. *Energy Sources Part A Recover. Util. Environ. Eff.* **2022**, 1–23. [CrossRef]
169. Altunay, H.C.; Albayrak, Z.; Özalp, A.N.; Çakmak, M. Analysis of anomaly detection approaches performed through deep learning methods in SCADA systems. In Proceedings of the 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 11–13 June 2021; pp. 1–6.
170. Ren, C.; Xu, Y. A fully data-driven method based on generative adversarial networks for power system dynamic security assessment with missing data. *IEEE Trans. Power Syst.* **2019**, *34*, 5044–5052. [CrossRef]
171. Liu, Y.; Ye, T.; Zeng, Z.; Zhang, Y.; Wang, G.; Chen, N.; Mao, C.; Yuan, X. Generative adversarial network-enabled learning scheme for power grid vulnerability analysis. *Int. J. Web Grid Serv.* **2021**, *17*, 138–151. [CrossRef]
172. Wang, C.; Sharifnia, E.; Gao, Z.; Tindemans, S.H.; Palensky, P. Generating multivariate load states using a conditional variational autoencoder. *Electr. Power Syst. Res.* **2022**, *213*, 108603. [CrossRef]
173. Gong, X.; Tang, B.; Zhu, R.; Liao, W.; Song, L. Data augmentation for electricity theft detection using conditional variational auto-encoder. *Energies* **2020**, *13*, 4291. [CrossRef]

174. Asre, S.; Anwar, A. Synthetic energy data generation using time variant generative adversarial network. *Electronics* **2022**, *11*, 355. [CrossRef]
175. Wang, Z.; Hong, T. Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN). *Energy Build.* **2020**, *224*, 110299. [CrossRef]
176. Zheng, X.; Wang, B.; Xie, L. Synthetic dynamic PMU data generation: A generative adversarial network approach. In Proceedings of the 2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA), College Station, TX, USA, 21–23 May 2019; pp. 1–6.
177. Langevin, A.; Carbonneau, M.A.; Cheriet, M.; Gagnon, G. Energy disaggregation using variational autoencoders. *Energy Build.* **2022**, *254*, 111623. [CrossRef]
178. Zhang, Z.; Wang, S.; Wang, P.; Jiang, P.; Zhou, H. Research on Fault Early Warning of Wind Turbine Based on IPSO-DBN. *Energies* **2022**, *15*, 9072. [CrossRef]
179. Li, B.; Li, Y.; Liang, C.; Su, W.; XuanYuan, Z. Data Augmentation and Class Based Model Evaluation for Load Disaggregation Based on Deep Learning. In *Proceedings of the 9th Frontier Academic Forum of Electrical Engineering: Volume II*; Springer: Singapore, 2021; pp. 331–346.
180. Kaselimi, M.; Voulodimos, A.; Protopapadakis, E.; Doulamis, N.; Doulamis, A. Energan: A generative adversarial network for energy disaggregation. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 1578–1582.
181. Khodayar, M.; Liu, G.; Wang, J.; Kaynak, O.; Khodayar, M.E. Spatiotemporal behind-the-meter load and PV power forecasting via deep graph dictionary learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4713–4727. [CrossRef]
182. Regan, J.; Saffari, M.; Khodayar, M. Deep attention and generative neural networks for nonintrusive load monitoring. *Electr. J.* **2022**, *35*, 107127. [CrossRef]
183. Jalali, S.M.J.; Khodayar, M.; Ahmadian, S.; Shafie-Khah, M.; Khosravi, A.; Islam, S.M.S.; Nahavandi, S.; Catalão, J.P. A new ensemble reinforcement learning strategy for solar irradiance forecasting using deep optimized convolutional neural network models. In Proceedings of the 2021 International Conference on Smart Energy Systems and Technologies (SEST), Vaasa, Finland, 6–8 September 2021; pp. 1–6.
184. Alobaidi, A.H.; Fazlhashemi, S.S.; Khodayar, M.; Wang, J.; Khodayar, M.E. Distribution Service Restoration with Renewable Energy Sources: A Review. *IEEE Trans. Sustain. Energy* **2022**, *14*, 1151–1168. [CrossRef]
185. Jalali, S.M.J.; Ahmadian, S.; Nakisa, B.; Khodayar, M.; Khosravi, A.; Nahavandi, S.; Islam, S.M.S.; Shafie-khah, M.; Catalão, J.P. Solar irradiance forecasting using a novel hybrid deep ensemble reinforcement learning algorithm. *Sustain. Energy Grids Netw.* **2022**, *32*, 100903. [CrossRef]
186. Zhang, Y.; Wang, X.; Wang, J.; Zhang, Y. Deep reinforcement learning based volt-var optimization in smart distribution systems. *IEEE Trans. Smart Grid* **2020**, *12*, 361–371. [CrossRef]
187. Cao, D.; Zhao, J.; Hu, W.; Ding, F.; Huang, Q.; Chen, Z.; Blaabjerg, F. Data-driven multi-agent deep reinforcement learning for distribution system decentralized voltage control with high penetration of PVs. *IEEE Trans. Smart Grid* **2021**, *12*, 4137–4150. [CrossRef]
188. Vu, T.L.; Mukherjee, S.; Yin, T.; Huang, R.; Tan, J.; Huang, Q. Safe reinforcement learning for emergency load shedding of power systems. In Proceedings of the 2021 IEEE Power & Energy Society General Meeting (PESGM), Washington, DC, USA, 26–29 July 2021; pp. 1–5.
189. Du, G.; Zou, Y.; Zhang, X.; Guo, L.; Guo, N. Heuristic energy management strategy of hybrid electric vehicle based on deep reinforcement learning with accelerated gradient optimization. *IEEE Trans. Transp. Electrif.* **2021**, *7*, 2194–2208. [CrossRef]
190. Zhang, H.; Peng, J.; Tan, H.; Dong, H.; Ding, F. A deep reinforcement learning-based energy management framework with lagrangian relaxation for plug-in hybrid electric vehicle. *IEEE Trans. Transp. Electrif.* **2020**, *7*, 1146–1160. [CrossRef]
191. Deltetto, D.; Coraci, D.; Pinto, G.; Piscitelli, M.S.; Capozzoli, A. Exploring the potentialities of deep reinforcement learning for incentive-based demand response in a cluster of small commercial buildings. *Energies* **2021**, *14*, 2933. [CrossRef]
192. Zhang, Y.; Ai, Q.; Li, Z. Intelligent demand response resource trading using deep reinforcement learning. *CSEE J. Power Energy Syst.* **2021**, *early access*.
193. Chen, T.; Bu, S.; Liu, X.; Kang, J.; Yu, F.R.; Han, Z. Peer-to-peer energy trading and energy conversion in interconnected multi-energy microgrids using multi-agent deep reinforcement learning. *IEEE Trans. Smart Grid* **2021**, *13*, 715–727. [CrossRef]
194. Zhang, G.; Hu, W.; Cao, D.; Liu, W.; Huang, R.; Huang, Q.; Chen, Z.; Blaabjerg, F. Data-driven optimal energy management for a wind-solar-diesel-battery-reverse osmosis hybrid energy system using a deep reinforcement learning approach. *Energy Convers. Manag.* **2021**, *227*, 113608. [CrossRef]
195. Yang, T.; Zhao, L.; Li, W.; Zomaya, A.Y. Dynamic energy dispatch strategy for integrated energy system based on improved deep reinforcement learning. *Energy* **2021**, *235*, 121377. [CrossRef]
196. Haque, N.I.; Shahriar, M.H.; Dastgir, M.G.; Debnath, A.; Parvez, I.; Sarwat, A.; Rahman, M.A. Machine learning in generation, detection, and mitigation of cyberattacks in smart grid: A survey. *arXiv* **2020**, arXiv:2010.00661.
197. Kamruzzaman, M.; Duan, J.; Shi, D.; Benidris, M. A deep reinforcement learning-based multi-agent framework to enhance power system resilience using shunt resources. *IEEE Trans. Power Syst.* **2021**, *36*, 5525–5536. [CrossRef]
198. Cao, D.; Hu, W.; Zhao, J.; Huang, Q.; Chen, Z.; Blaabjerg, F. A multi-agent deep reinforcement learning based voltage regulation using coordinated PV inverters. *IEEE Trans. Power Syst.* **2020**, *35*, 4120–4123. [CrossRef]

199. Diao, R.; Wang, Z.; Shi, D.; Chang, Q.; Duan, J.; Zhang, X. Autonomous voltage control for grid operation using deep reinforcement learning. In Proceedings of the 2019 IEEE Power & Energy Society General Meeting (PESGM), Atlanta, GA, USA, 4–8 August 2019; pp. 1–5.
200. Wang, S.; Duan, J.; Shi, D.; Xu, C.; Li, H.; Diao, R.; Wang, Z. A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning. *IEEE Trans. Power Syst.* **2020**, *35*, 4644–4654. [CrossRef]
201. Vu, T.L.; Mukherjee, S.; Huang, R.; Huang, Q. Barrier function-based safe reinforcement learning for emergency control of power systems. In Proceedings of the 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, 14–17 December 2021; pp. 3652–3657.
202. Li, X.; Wang, X.; Zheng, X.; Dai, Y.; Yu, Z.; Zhang, J.J.; Bu, G.; Wang, F.Y. Supervised assisted deep reinforcement learning for emergency voltage control of power systems. *Neurocomputing* **2022**, *475*, 69–79. [CrossRef]
203. Dai, Y.; Chen, Q.; Zhang, J.; Wang, X.; Chen, Y.; Gao, T.; Xu, P.; Chen, S.; Liao, S.; Jiang, H.; et al. Enhanced oblique decision tree enabled policy extraction for deep reinforcement learning in power system emergency control. *Electr. Power Syst. Res.* **2022**, *209*, 107932. [CrossRef]
204. Zhang, K.; Zhang, J.; Xu, P.D.; Gao, T.; Gao, D.W. Explainable AI in deep reinforcement learning models for power system emergency control. *IEEE Trans. Comput. Soc. Syst.* **2021**, *9*, 419–427. [CrossRef]
205. Chen, J.; Shu, H.; Tang, X.; Liu, T.; Wang, W. Deep reinforcement learning-based multi-objective control of hybrid power system combined with road recognition under time-varying environment. *Energy* **2022**, *239*, 122123. [CrossRef]
206. Qian, T.; Shao, C.; Li, X.; Wang, X.; Chen, Z.; Shahidehpour, M. Multi-agent deep reinforcement learning method for EV charging station game. *IEEE Trans. Power Syst.* **2021**, *37*, 1682–1694. [CrossRef]
207. Hasanvand, S.; Rafiei, M.; Gheisarnejad, M.; Khooban, M.H. Reliable power scheduling of an emission-free ship: Multiobjective deep reinforcement learning. *IEEE Trans. Transp. Electrif.* **2020**, *6*, 832–843. [CrossRef]
208. Qian, T.; Shao, C.; Wang, X.; Shahidehpour, M. Deep reinforcement learning for EV charging navigation by coordinating smart grid and intelligent transportation system. *IEEE Trans. Smart Grid* **2019**, *11*, 1714–1723. [CrossRef]
209. Pallonetto, F.; Jin, C.; Mangina, E. Forecast electricity demand in commercial building with machine learning models to enable demand response programs. *Energy AI* **2022**, *7*, 100121. [CrossRef]
210. Bahrami, S.; Chen, Y.C.; Wong, V.W. Deep reinforcement learning for demand response in distribution networks. *IEEE Trans. Smart Grid* **2020**, *12*, 1496–1506. [CrossRef]
211. Lu, R.; Li, Y.C.; Li, Y.; Jiang, J.; Ding, Y. Multi-agent deep reinforcement learning based demand response for discrete manufacturing systems energy management. *Appl. Energy* **2020**, *276*, 115473. [CrossRef]
212. Wen, L.; Zhou, K.; Li, J.; Wang, S. Modified deep learning and reinforcement learning for an incentive-based demand response model. *Energy* **2020**, *205*, 118019. [CrossRef]
213. Zhong, S.; Wang, X.; Zhao, J.; Li, W.; Li, H.; Wang, Y.; Deng, S.; Zhu, J. Deep reinforcement learning framework for dynamic pricing demand response of regenerative electric heating. *Appl. Energy* **2021**, *288*, 116623. [CrossRef]
214. Cao, J.; Harrold, D.; Fan, Z.; Morstyn, T.; Healey, D.; Li, K. Deep reinforcement learning-based energy storage arbitrage with accurate lithium-ion battery degradation model. *IEEE Trans. Smart Grid* **2020**, *11*, 4513–4521. [CrossRef]
215. Liang, Y.; Guo, C.; Ding, Z.; Hua, H. Agent-based modeling in electricity market using deep deterministic policy gradient algorithm. *IEEE Trans. Power Syst.* **2020**, *35*, 4180–4192. [CrossRef]
216. Ye, Y.; Qiu, D.; Li, J.; Strbac, G. Multi-period and multi-spatial equilibrium analysis in imperfect electricity markets: A novel multi-agent deep reinforcement learning approach. *IEEE Access* **2019**, *7*, 130515–130529. [CrossRef]
217. Chung, H.M.; Maharjan, S.; Zhang, Y.; Eliassen, F. Distributed deep reinforcement learning for intelligent load scheduling in residential smart grids. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2752–2763. [CrossRef]
218. Li, Y.; Wang, R.; Yang, Z. Optimal scheduling of isolated microgrids using automated reinforcement learning-based multi-period forecasting. *IEEE Trans. Sustain. Energy* **2021**, *13*, 159–169. [CrossRef]
219. Huang, B.; Wang, J. Deep-reinforcement-learning-based capacity scheduling for PV-battery storage system. *IEEE Trans. Smart Grid* **2020**, *12*, 2272–2283. [CrossRef]
220. Liang, Y.; Ding, Z.; Ding, T.; Lee, W.J. Mobility-aware charging scheduling for shared on-demand electric vehicle fleet using deep reinforcement learning. *IEEE Trans. Smart Grid* **2020**, *12*, 1380–1393. [CrossRef]
221. Liu, X.; Ospina, J.; Konstantinou, C. Deep reinforcement learning for cybersecurity assessment of wind integrated power systems. *IEEE Access* **2020**, *8*, 208378–208394. [CrossRef]
222. Karimipour, H.; Dehghantanha, A.; Parizi, R.M.; Choo, K.K.R.; Leung, H. A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids. *IEEE Access* **2019**, *7*, 80778–80788. [CrossRef]
223. Li, Y.; Wu, J. Low latency cyberattack detection in smart grids with deep reinforcement learning. *Int. J. Electr. Power Energy Syst.* **2022**, *142*, 108265. [CrossRef]
224. Bârgăuan, B.; Creţu, M.; Fati, O.; Ceclan, A.; Darabant, L.; Micu, D.D.; Şteţ, D.; Czumbil, L. Energy Management System for the Demand Response in TUCN Buildings. In Proceedings of the 2018 53rd International Universities Power Engineering Conference (UPEC), Glasgow, UK, 4–7 September 2018; pp. 1–4. [CrossRef]
225. Damjanović, I.; Pavić, I.; Puljiz, M.; Brcic, M. Deep Reinforcement Learning-Based Approach for Autonomous Power Flow Control Using Only Topology Changes. *Energies* **2022**, *15*, 6920. [CrossRef]

226. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, T.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Curran Associates, Inc., Long Beach, CA, USA, 4–9 December 2017; Volume 30.
227. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
228. Moritz, N.; Hori, T.; Le, J. Streaming Automatic Speech Recognition with the Transformer Model. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6074–6078. ISSN: 2379-190X. [CrossRef]
229. Mott, A.; Zoran, D.; Chrzanowski, M.; Wierstra, D.; Jimenez Rezende, D. Towards Interpretable Reinforcement Learning Using Attention Augmented Agents. In Proceedings of the Advances in Neural Information Processing Systems, Curran Associates, Inc., Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
230. Xie, B.; Li, S.; Lv, F.; Liu, C.H.; Wang, G.; Wu, D. A Collaborative Alignment Framework of Transferable Knowledge Extraction for Unsupervised Domain Adaptation. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 6518–6533. [CrossRef]
231. Bouteldja, N.; Klinkhammer, B.M.; Schlaich, T.; Boor, P.; Merhof, D. Improving unsupervised stain-to-stain translation using self-supervision and meta-learning. *J. Pathol. Inform.* **2022**, *13*, 100107. [CrossRef]
232. Zhu, Y.; Zhuang, F.; Wang, J.; Ke, G.; Chen, J.; Bian, J.; Xiong, H.; He, Q. Deep Subdomain Adaptation Network for Image Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 1713–1722. [CrossRef] [PubMed]
233. Liu, X.; He, P.; Chen, W.; Gao, J. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Florence, Italy, 2019; pp. 4487–4496. [CrossRef]
234. Bell, P.; Fainberg, J.; Klejch, O.; Li, J.; Renals, S.; Swietojanski, P. Adaptation Algorithms for Neural Network-Based Speech Recognition: An Overview. *IEEE Open J. Signal Process.* **2021**, *2*, 33–66. [CrossRef]
235. Tsang, M.; Rambhatla, S.; Liu, Y. How does This Interaction Affect Me? Interpretable Attribution for Feature Interactions. In Proceedings of the Advances in Neural Information Processing Systems, Curran Associates, Inc., virtual, 6–12 December 2020; Volume 33, pp. 6147–6159.
236. Panwar, H.; Gupta, P.K.; Siddiqui, M.K.; Morales-Menendez, R.; Bhardwaj, P.; Singh, V. A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. *Chaos Solitons Fractals* **2020**, *140*, 110190. [CrossRef]
237. Mahapatra, D.; Ge, Z.; Reyes, M. Self-Supervised Generalized Zero Shot Learning for Medical Image Classification Using Novel Interpretable Saliency Maps. *IEEE Trans. Med. Imaging* **2022**, *41*, 2443–2456. [CrossRef] [PubMed]
238. Bass, C.; da Silva, M.; Sudre, C.; Tudosiu, P.D.; Smith, S.; Robinson, E. ICAM: Interpretable Classification via Disentangled Representations and Feature Attribution Mapping. In Proceedings of the Advances in Neural Information Processing Systems, virtual, 6–12 December 2020; Curran Associates, Inc.: Red Hook, NY, USA; Volume 33, pp. 7697–7709.
239. Wang, Z.; Zhang, W.; Liu, N.; Wang, J. Scalable Rule-Based Representation Learning for Interpretable Classification. In Proceedings of the Advances in Neural Information Processing Systems, virtual, 6–14 December 2021; Curran Associates, Inc.: Red Hook, NY, USA; Volume 34, pp. 30479–30491.
240. Toğaçar, M.; Muzoğlu, N.; Ergen, B.; Yarman, B.S.B.; Halefoğlu, A.M. Detection of COVID-19 findings by the local interpretable model-agnostic explanations method of types-based activations extracted from CNNs. *Biomed. Signal Process. Control.* **2022**, *71*, 103128. [CrossRef] [PubMed]
241. Antwarg, L.; Miller, R.M.; Shapira, B.; Rokach, L. Explaining anomalies detected by autoencoders using Shapley Additive Explanations. *Expert Syst. Appl.* **2021**, *186*, 115736. [CrossRef]
242. Vinuesa, R.; Sirmacek, B. Interpretable deep-learning models to help achieve the Sustainable Development Goals. *Nat. Mach. Intell.* **2021**, *3*, 926. [CrossRef]
243. Salahuddin, Z.; Woodruff, H.C.; Chatterjee, A.; Lambin, P. Transparency of deep neural networks for medical image analysis: A review of interpretability methods. *Comput. Biol. Med.* **2022**, *140*, 105111. [CrossRef]
244. Wang, F.; Kaushal, R.; Khullar, D. Should Health Care Demand Interpretable Artificial Intelligence or Accept "Black Box" Medicine? *Ann. Intern. Med.* **2020**, *172*, 59–60. [CrossRef]
245. Mullins, M.; Holland, C.P.; Cunneen, M. Creating ethics guidelines for artificial intelligence and big data analytics customers: The case of the consumer European insurance market. *Patterns* **2021**, *2*, 100362. [CrossRef]
246. Wang, J.; Li, Y.; Zhao, R.; Gao, R.X. Physics guided neural network for machining tool wear prediction. *J. Manuf. Syst.* **2020**, *57*, 298–310. [CrossRef]
247. Yan, W.; Melville, J.; Yadav, V.; Everett, K.; Yang, L.; Kesler, M.S.; Krause, A.R.; Tonks, M.R.; Harley, J.B. A novel physics-regularized interpretable machine learning model for grain growth. *Mater. Des.* **2022**, *222*, 111032. [CrossRef]
248. Hu, X.; Hu, H.; Verma, S.; Zhang, Z.L. Physics-Guided Deep Neural Networks for Power Flow Analysis. *IEEE Trans. Power Syst.* **2021**, *36*, 2082–2092. [CrossRef]
249. Almajid, M.M.; Abu-Al-Saud, M.O. Prediction of porous media fluid flow using physics informed neural networks. *J. Pet. Sci. Eng.* **2022**, *208*, 109205. [CrossRef]
250. He, Z.; Ni, F.; Wang, W.; Zhang, J. A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials. *Mater. Today Commun.* **2021**, *28*, 102719. [CrossRef]

251. Zhang, D.; Liu, X.; Xia, J.; Gao, Z.; Zhang, H.; de Albuquerque, V.H.C. A Physics-guided Deep Learning Approach For Functional Assessment of Cardiovascular Disease in IoT-based Smart Health. *IEEE Internet Things J.* **2023**, *early access*. [CrossRef]
252. Kashinath, K.; Mustafa, M.; Albert, A.; Wu, J.L.; Jiang, C.; Esmaeilzadeh, S.; Azizzadenesheli, K.; Wang, R.; Chattopadhyay, A.; Singh, A.; et al. Physics-informed machine learning: Case studies for weather and climate modelling. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2021**, *379*, 20200093. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.