electronics

MDPI

Review

A Study of Optimization in Deep Neural Networks for Regression

Chieh-Huang Chen [1], Jung-Pin Lai [2], Yu-Ming Chang [3], Chi-Ju Lai [4] and Ping-Feng Pai [1,4,*]

[1] PhD Program in Strategy and Development of Emerging Industries, National Chi Nan University, Nantou 54561, Taiwan; s106245911@ncnu.edu.tw
[2] Department of Multimedia Game Development and Application, HungKuang University, Taichung 43302, Taiwan; jungpin.lai@sunrise.hk.edu.tw
[3] Department of Culinary Arts and Hotel Management, HungKuang University, Taichung 43302, Taiwan; right@sunrise.hk.edu.tw
[4] Department of Information Management, National Chi Nan University, Nantou 54561, Taiwan; s107213029@ncnu.edu.tw
* Correspondence: paipf@ncnu.edu.tw

Abstract: Due to rapid development in information technology in both hardware and software, deep neural networks for regression have become widely used in many fields. The optimization of deep neural networks for regression (DNNR), including selections of data preprocessing, network architectures, optimizers, and hyperparameters, greatly influence the performance of regression tasks. Thus, this study aimed to collect and analyze the recent literature surrounding DNNR from the aspect of optimization. In addition, various platforms used for conducting DNNR models were investigated. This study has a number of contributions. First, it provides sections for the optimization of DNNR models. Then, elements of the optimization of each section are listed and analyzed. Furthermore, this study delivers insights and critical issues related to DNNR optimization. Optimizing elements of sections simultaneously instead of individually or sequentially could improve the performance of DNNR models. Finally, possible and potential directions for future study are provided.

Keywords: optimization; deep neural networks; regression

check for updates

Citation: Chen, C.-H.; Lai, J.-P.; Chang, Y.-M.; Lai, C.-J.; Pai, P.-F. A Study of Optimization in Deep Neural Networks for Regression. *Electronics* **2023**, *12*, 3071. https://doi.org/10.3390/electronics12143071

Academic Editors: Juan M. Corchado, Carlos A. Iglesias, Byung-Gyu Kim, Rashid Mehmood, Fuji Ren and In Lee

Received: 16 June 2023
Revised: 11 July 2023
Accepted: 13 July 2023
Published: 14 July 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, owing to the rapid advancements in computing power, machine learning and deep learning have gained widespread attention. In particular, given a much stronger capability of learning essential features from big data, deep learning models cannot only achieve better performance than traditional machine learning models but can also be utilized to cope with more complex problems, including clustering, classification, and regression [1,2].

Deep neural networks for regression (DNNR) have been broadly employed in various fields, such as climate [3], agricultural planting [4], and renewable energy forecasting [5]. Several studies have revealed that adjusting the models' hyperparameters through algorithms can achieve better performance. Xiong et al. [6] proposed an ensemble deep learning model that combined the ant colony optimization (ACO) strategy and deep belief networks to solve the problem of landslide susceptibility mapping. The numerical results illustrated that the ensemble model could provide better performance due to the use of ACO for optimizing multiple parameters simultaneously. Liu et al. [7] applied LSTM networks to forecast subway passenger flows in China. Two months of automated fare collection data from Beijing Metro's Xizhimen Station and particle swarm optimization were used to train and optimize the forecasting models. Compared with the support vector regression model, the proposed optimized-parameter model could achieve better performance. Gao et al. [8] designed LSTM and GRU models to forecast stock prices. Least absolute shrinkage and

selection operator (LASSO) and principal component analysis (PCA) methods were, respectively, employed to perform data preprocessing of dimension reduction. The experimental results indicated that the proposed model could obtain satisfactory performance. Parameter tuning plays a critical role in achieving better performance and shortening the computing time of DNNR models [9].

Optimization is a crucial issue of DNNR models, as it directly influences their performance and computing time. Some studies have dedicated considerable efforts to devising approaches for optimizing DNNR models. Dong et al. [10] depicted a number of challenges related to DNNR models, including optimizing hyperparameters and preventing overfitting. The proper determination of learning and dropout rates can improve the performance of DNNR models. How to reduce the training time caused by constantly adjusting parameters is a critical issue. However, resolutions for this issue have not been provided. Additionally, studies have indicated that the optimization of DNNR models is a non-deterministic polynomial-time hardness problem [10,11]. Wang et al. [5] reviewed the optimization of DNNR models in renewable energy forecasting. The sections of optimization included data preprocessing, architectures, and hyperparameters. This study pointed out that data preprocessing can decrease the interference when training DNNR models and therefore improve the performance of the regression. In addition, the design of architectures and the setting of hyperparameters can effectively prevent the value of loss function from falling into local minima. Trial-and-error and metaheuristics were employed to optimize three sections of DNNR models. Compared with Dong et al. [10], this study presented various metaheuristics to optimize these parameters but there was no further explanation of how to implement them. Abd Elaziz et al. [11] investigated metaheuristics including swarm intelligence, evolutionary computing, natural phenomena, and human inspiration for optimizing DNNR models. Metaheuristics introduced in this study were used for optimizing various network models. In particular, the various combinations of hyperparameters directly affect the performance of DNNR models. Akay et al. [12] provided an overview of DNNR optimization problems, including architecture optimization, hyperparameter optimization, training, and feature selection. This study illustrated that simplifying the network architectures and data sizes are feasible alternatives to reduce the training time. In addition, this investigation explored the use of metaheuristics in DNNR models and pointed out future directions for their integration. Zhan et al. [13] provided more explicit discussions and explanations of the hyperparameters for each training stage of a DNNR model, namely, data preprocessing, model searches, training, and evaluation. This study indicated that combining optimization sections is more suitable for optimizing DNNR models than only optimizing individual sections.

In the previous survey literature collected by this study, the limited literature discussed investigating the optimization of DNNR models in terms of three sections [5,13] and platforms [10]. Additionally, the impact of different platforms used for training the DNNR models has received limited attention in the literature. Moreover, the selection of optimizers has a significant influence on computational time. Therefore, the research gap arises in the following aspects. First, the optimization of four sections of DNNR models and platforms for developing DNNR models were investigated in this study. This study aims to depict the combinatorial optimization and various characteristics of platforms when developing DNNR models. Secondly, an analysis of the relationship between platforms and optimizers is presented. Lastly, to achieve optimal performance, this study delved into parameters in each optimization section and statistically providing appropriate settings. Table 1 illustrates a comparative analysis between this investigation and relevant survey studies, highlighting the specific optimization sections under scrutiny.

In this investigation, four deep neural networks for regression models (Shickel et al. [14]), namely, CNN, LSTM, GRU, and DBN, were employed to explore optimization tasks in DNNR. This study was conducted using the keywords “deep neural networks”, “regression”, and “optimization” to first search the literature published between 2019 and 2023. Then, the collected articles were organized based on the metadata of parameters used in

four optimization sections. The third step involved an in-depth analysis of the impact and challenges of parameters for model optimization, along with the optimization algorithms employed. Finally, the findings and future research directions were derived from the collected literature. Figure 1 presents the flowchart and four optimization sections for the DNNR models investigated in this study. The selection of data preprocessing procedures is dependent on data patterns and is interactively related to the other three optimization sections, namely, the architectures, optimizers, and hyperparameters of DNNR models. Data used for the DNNR models were divided into training data and testing data. The training data were used to model the DNNR, and the testing data were employed to examine the performance of the DNNR models. Then, optimization methods were used to determine the parameters of the architectures, optimizers, and hyperparameters for training the DNNR models. Well-trained DNNR models were obtained when the stop criteria were reached. Finally, the testing data were used to evaluate the performance of the DNNR models.

Table 1. A comparison with the related survey literature.

Literature	Platforms	Sections of Optimization
This study	1. TensorFlow with Keras 2. PyTorch 3. Caffe 4. Theano with Keras 5. Matlab	1. Data preprocessing 2. Architectures 3. Optimizers 4. Hyperparameters
[10]	TensorFlow	Hyperparameters
[5,13]	Unavailable	1. Data preprocessing 2. Architectures 3. Hyperparameters
[11,12]	Unavailable	1. Architectures 2. Hyperparameters

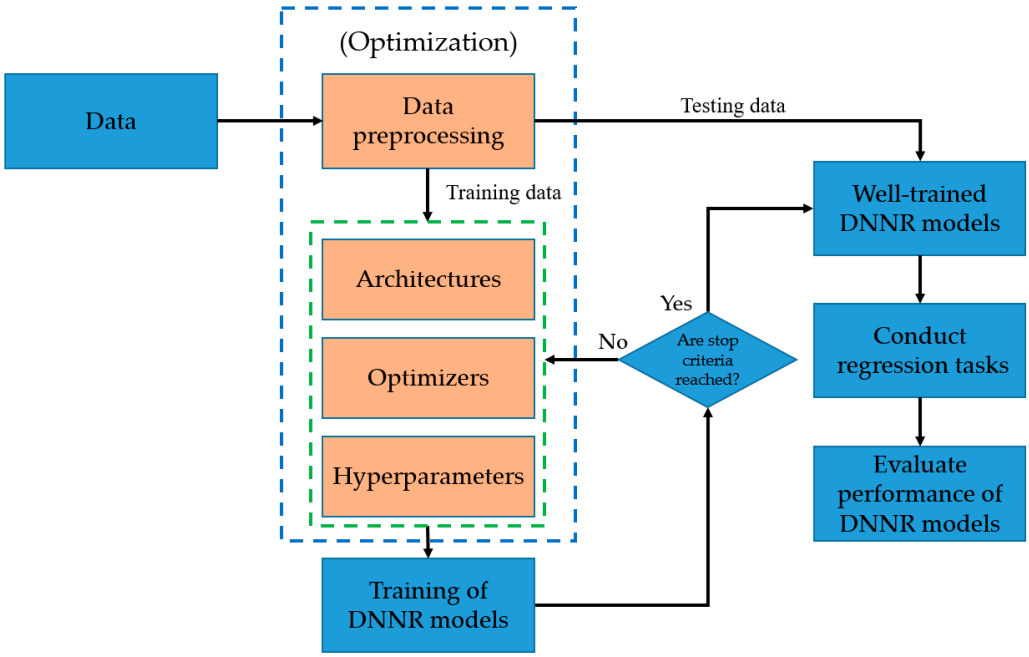

Figure 1. Optimization of deep neural networks for regression.

The rest of this study is organized as follows. Section 2 depicts four sections of optimization of the DNNR models. Section 3 illustrates the optimization of the DNNR

models, systematically and globally. Finally, conclusions and future work are addressed in Section 4.

2. Sections of Optimization of Deep Neural Networks for Regression

The performance and accuracy of DNNR models are highly influenced by the selection and combinations of parameters, such as data preprocessing, architectures, hyperparameters, and optimizers. Therefore, to improve the performance of DNNR models, combinatorically optimizing four sections at the same time is a challenging issue worthy of further investigation. Table 2 lists the metadata of parameters and data types based on four sections.

Table 2. Metadata of parameters.

Sections	Parameters	Data Types
Architectures	Number of layers	Integer
	Number of nodes	Integer
	Number of kernel sizes	Integer
	Number of pooling	Integer
Optimizers	Optimizers	Categories (Adadelta, Adagrad, Adam, Adamx, Ftrl, Nadam, RMSprop, SGD, SparseAdam, Adamax, ASGD, LBFGS, NAdam, RAdam, Rprop, Nesterov)
Hyperparameters	Learning rates	Real
	Number of epochs	Integer
	Batch sizes	Integer
	Iterations	Integer
	Dropout rates	Real
Data preprocessing	Tasks	Categories (missing values, dimensionality reduction, removing outliers, feature selection, decomposition, normalization)

2.1. Data Preprocessing

Data preprocessing is highly related to the effectiveness and efficiency of DNNR models and thus is an essential factor in searching for proper hyperparameters for DNNR models [5,15]. Table 3 lists six data preprocessing methods commonly used in recent studies. It can be observed that normalization is one of the most frequently used data preprocessing methods for DNNR models.

Table 3. A list of data preprocessing methods.

Tasks of Data Preprocessing	Methods
Missing values	Supplementing average value [16,17], linear interpolation [17,18], KNN [19,20]
Dimensionality reduction	PCA [8], pooling layer [21], t-SNE [22], SPCA [23]
Removing outliers	Pauta criterion [18], EWMA [24]
Feature selection	PSO [1], LASSO [8,25], ASO [26], GA [27], MI [28], GRA [29], PCC [30], CCA [31]
Decomposition	EMD [32], EEMD [33], CEEMDAN [19,27,34–38], ICEEMDAN [39], SSA [40,41], VMD [42], SVMD [43]
Normalization	[6,17,20,27,30,31,34,42,44–55]

Note: KNN = k-nearest neighbor algorithm; PCA = principal component analysis; t-SNE = t-distributed stochastic neighbor embedding; SPCA = sparse principal component analysis; EWMA = exponentially weighted moving average; PSO = particle swarm optimization; LASSO = least absolute shrinkage and selection operator; ASO = atom search optimization; GA = genetic algorithm; MI = mutual-information-based; GRA = grey relational analysis; PCC = Pearson correlation coefficient; CCA = correlation coefficient analysis; EMD = empirical mode decomposition; EEMD = ensemble empirical mode decomposition; CEEMDAN = complementary ensemble empirical mode decomposition with adaptive noise; ICEEMDAN = improved complementary ensemble empirical mode decomposition with adaptive noise; SSA = singular spectrum analysis; VMD = variational mode decomposition; SVMD = successive variational mode decomposition.

Before data can be used for training, it is necessary to ensure that no values are missing to reduce model bias or avoid presenting misleading analysis results. Many methods have been developed to cope with missing values, include supplementing the average value [16,17], linear interpolation [17,18], and machine learning methods [19,20]. Two methods, linear interpolation and average values, were used in the study by Tsokov et al. [17]. Linear interpolation and the average value are used when the gap is small and huge, respectively. The gap is mainly based on empirical values. Different from Tsokov et al. [17], Shao et al. [16,19] utilized the KNN method to obtain missing values by calculating the average value of adjacent data, which is not necessary to consider the gap of missing values.

High-dimensional data can easily cause the model to become diverse and increases the modeling time. Consequently, data dimension reduction is necessary. Dimensionality reduction only transforms features into a lower dimension, which can help to identify the features that should be used in the modeling and avoid interfering with the model's failure to converge. The pooling layer of the convolutional neural network can reduce the dimensionality of the data [21]. Other methods, including principal component analysis [8], t-distributed stochastic neighbor embedding [22], and sparse principal component analysis [23], can also be used to reduce dimensions. Feature selection is similar to dimensionality reduction, but without changing the data. Zhao et al. [25] used the least absolute shrinkage and selection operator method to select important variables for modeling, which increased the modeling efficiency and performance.

Outliers are generally removed to exclude abnormal or anomalous data [18,24] and improve data quality. In the study of Cheng et al. [24], the exponentially weighted moving average was employed to smooth and remove noise data. The experimental results showed that the proposed models could perform better than without removing outliers. Decomposition methods are used to process datasets, such as network traffic datasets [19], solar datasets [34], and wind datasets [39], to avoid noise in model distortion. Li et al. [32] compared the performance of DNNR models using empirical mode decomposition and those without it. The experimental results proved that models with decomposition methods have better accuracy.

2.2. Network Architecture Selection

The architecture of a DNNR model contains an input layer, an output layer, and hidden layers. Each layer includes many nodes. The optimization of the architecture is based on determining the number of layers and nodes of the DNNR model [56]. Table 4 lists the methods, DNNR models, and parameters of the architecture for optimization. Network architecture selection can be divided into optimal tasks and trial-and-error methods. Most studies fix the number of layers and only optimize the number of nodes; few investigations have considered both the number of layers and nodes at the same time.

It is difficult for DNNR models to generate proper parameters within an acceptable training period when the data amount is enormous [77]; however, a number of optimal methods have been employed to solve this problem, including grid searches, Bayesian methods, and metaheuristics. Sen et al. [57] used a grid search to find the number of nodes in each layer and the number of layers in CNN and LSTM networks when designing the network architecture. Yang and Liu [35] used an improved whale optimization algorithm to optimize the number of layers and nodes in the GRU algorithm for water quality prediction. The results showed that the proposed model could outperform the original GRU model. Chen et al. [55] used the particle swarm optimization algorithm to simultaneously search for the number of hidden layers and nodes. The experimental results pointed out that the optimized LSTM model could perform better than the standard LSTM model.

Table 4. Architecture selection for DNNR models in terms of optimization methods.

Models	CNN			LSTM/GRU/DBN	
Parameters / Methods	Number of Layers, Kernel Sizes, and Pooling	Number of Kernel Sizes	Number of Layers and Nodes	Number of Nodes	Number of Layers
Grid search	[57,58]	[34]		[30]	
Bayesian	[59]			[2,42,53,60,61]	
GA	[17]				
IBFO			[46]		
ABC	[62]				
SOA				[33,43]	
IWOA			[35]	[38,63,64]	
IGOA				[37]	
SCA				[65]	
GWO	[66]			[67]	
DEA			[68]		
PSO/IPSO SPSO/SAPSO	[69]	[21]	[55]	[7,19,31,49,52,70,71]	[72]
tSBO		[25]			
SSA			[73]	[74]	
FWA			[75]		
BER			[76]		
Trial-and-error	[77]		[1,18,48,78–80]	[27,81,82]	

Note: GA = genetic algorithm; IBFO = intelligent bacterial foraging optimization; ABC = artificial bee colony; SOA = seeker optimization algorithm; IWOA = improved whale optimization algorithm; IGOA = improved grasshopper optimization algorithm; SCA = the sine cosine algorithm; GWO = gray wolf optimizer; DEA = differential evolution algorithm; PSO = particle swarm optimization; IPSO = improved particle swarm optimization; SPSO = selective particle swarm optimization; SAPSO = simulated annealing particle swarm optimization; tSBO = T-distribution stain bower bird optimization algorithms; SSA = sparrow search algorithm; FWA = fireworks algorithm; BER = Al-Biruni Earth radius.

When the trial-and-error technique is used to select suitable network architectures, more time is generally required for the search. Many researchers have determined the network layer architecture based on their experience and then used optimization algorithms to determine the number of nodes of each hidden layer [18,78–80]. Ghimire et al. [1] indicated that an insufficient number of hidden layers could cause the DBN to fail to obtain enough feature space and cause the model to be under-fitted, but that having too many hidden layers could lead to overfitting. Song et al. [30] used a fixed two-layer GRU and defined the initial learning rates, number of hidden nodes, step sizes, batch sizes, and iterations, by experience. Gao et al. [48] used GRU to study short-term runoff predictions, and used the trial-and-error method to determine the number of layers. This study reported that a GRU with a single hidden layer with 20 nodes could outperform a GRU with many hidden layers. Trial-and-error methods are not efficient and effective ways of optimizing DNNR models due to the impossibility of enumerating all combinations of parameters.

Determining the network architecture by trial-and-error is a time-consuming task that cannot easily reach an optimal or near-optimal combination of layers and nodes, and it is nearly impossible to enumerate all of the combinations. Therefore, most studies set the number of network layers first and then use optimization algorithms to search the appropriate number of nodes for each layer.

Uzair and Jamil [83] showed that the use of three hidden layers can be used with less training, and higher accuracy can be achieved in an acceptable time. When the number of hidden layers is more than three, the training time increases dramatically, but only a limited improvement in accuracy is obtained. Therefore, for DNNR models, three hidden layers are recommended for the initial setting value, and after which, optimization methods should be used to select the number of nodes for each hidden layer.

2.3. Selection of Optimizers

Optimizers represent a crucial parameter within a neural network, as they can adjust weights and biases continuously to achieve the least value in the loss function. Table 5 shows statistics about the optimizers used in the related literature. The Adam optimizer, a stochastic gradient descent method derived from RMSProp [48,84], is extensively employed in most studies, as illustrated in Figure 2. Moreover, the Adam algorithm has the capability to compute an adaptive learning rate for each parameter, enabling the model to accelerate convergence and minimize fluctuations. The high computational efficiency and low memory storage have positioned the Adam algorithm as one of the most frequently and popularly utilized optimizers [33,51,85,86].

Table 5. Optimizers.

DNNR Models	Adam (Admax)	RMSProp	Unavailable
CNN	[17,26,34,40,41,57,58,62,77,85,87–89]		[25,39,59,66,69,90,91]
LSTM	[2,7,19,31,52–54,60,61,63,68,71,75,81,86,92–94]		[8,23,37,38,43,55,65,67,74,76,95–97]
GRU	[18,24,30,33,46,49–51,78–80]	[48,84]	[36,98–100]
DBN	[6,20,101]		[1,28,32,42,44,45,64,70,72,73,82,102–105]

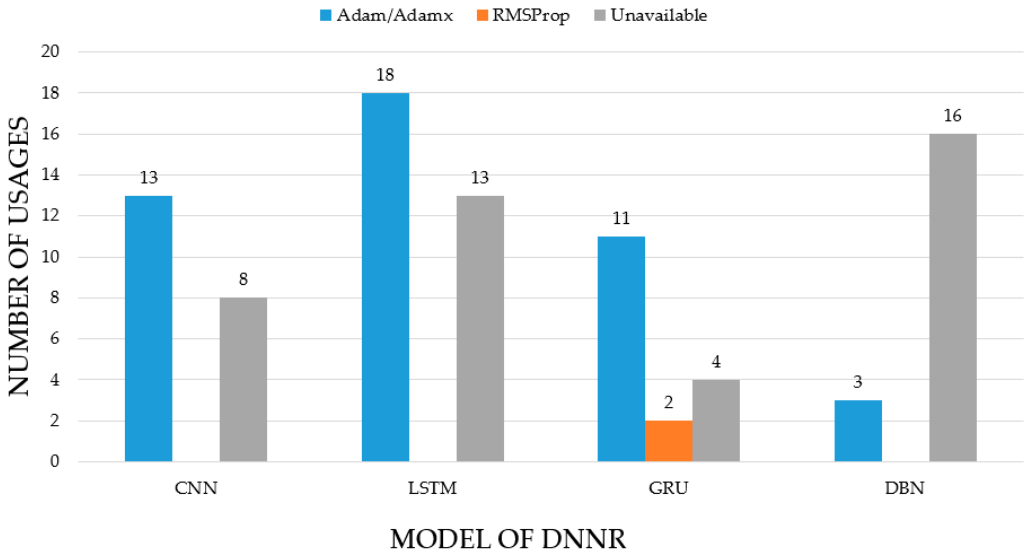

Figure 2. Optimizers used in DNNR models.

As shown in Figure 3, most studies used the TensorFlow platform to develop their work. TensorFlow provides a comprehensive development ecosystem, which not only includes graphical debugging tools but also offers an environment for transforming research projects into commercial applications. Although PyTorch owns more options for optimizers, the user-friendliness and extensive online resources make TensorFlow more commonly used for developing projects. The optimizers provided by deep learning platforms are listed in Table 6. In the current TensorFlow platform, eight standard optimizer algorithms are provided, and among which, Adam is the most commonly used optimizer.

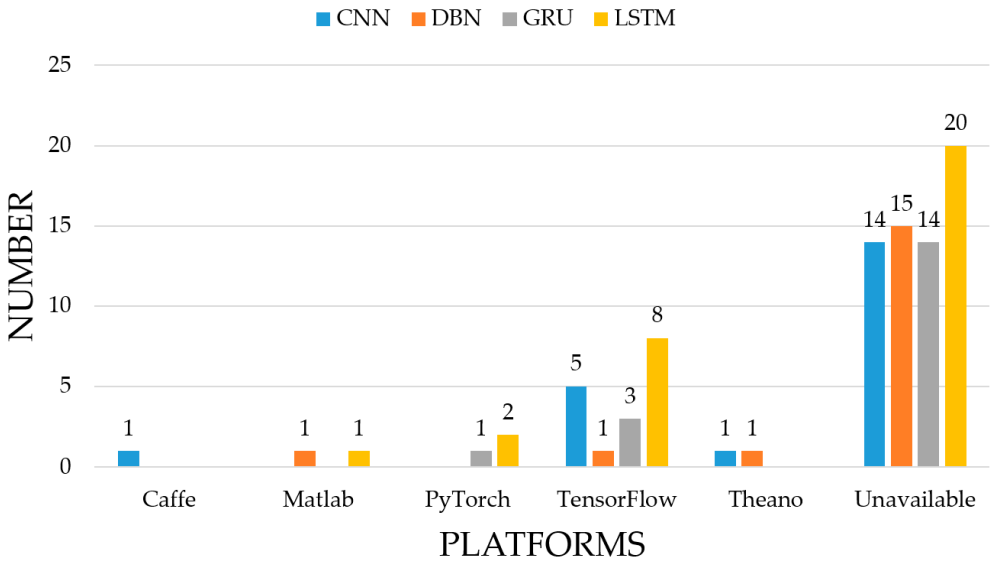

Figure 3. Platforms used for DNNR.

Table 6. Optimizers used for DNNR platforms.

Optimizers	Deep Learning Platforms				
	Caffe	**Matlab**	**PyTorch**	**TensorFlow with Keras**	**Theano with Keras**
Adadelta	X		X	X	X
Adagrad	X		X	X	X
Adam	X	X	X	X	X
Adamx			X	X	X
Ftrl				X	X
Nadam				X	X
RMSprop	X	X	X	X	X
SGD	X	X	X	X	X
SparseAdam			X		
Adamax			X		
ASGD			X		
LBFGS			X		
NAdam			X		
RAdam			X		
Rprop			X		
Nesterov	X				

Note: X = The platform contains the optimizer.

2.4. Hyperparameter Tuning

The fine-tuning of hyperparameters is essential for training a model and is closely related to the model's accuracy. Fine-tuning the learning rate, batch size, and iteration can prevent DNNR models from falling into the local optimum while reducing the modeling time and cost [10].

In the related research on network models, except for the unique hyperparameters of individual network models, there are five common hyperparameters, namely, the learning rate, epochs, batch size, iteration, and dropout rate. Among them, the learning rate is an essential parameter in most studies. It is one of the parameters of the optimizer, which is used to determine the speed of finding the optimal weight [58]. If the learning rate is too low, the modeling time will be longer; if the learning rate is too high, the model will fail to converge [63]. Wang et al. [44] used DBN to predict the survival of cancer patients, and the results showed that the learning rate had a high impact on the model's accuracy. Yalçın et al. [62] mentioned that setting a suitable learning rate can avoid overfitting.

In addition, other commonly used parameters are epoch, batch size, and iteration. Yu et al. [78] highlighted that the iteration affects whether the model becomes overtrained. The batch size is used in gradient descent to control the number of training samples used to update the internal parameters of a network model. Wang et al. [44] showed that the batch size significantly impacts the accuracy in the DBN model. Iteration and epoch are parameters related to batch size optimization. The equation is Iteration = (Dataset size/Batch size) × Epoch. These parameters are related to the amount of training data and will affect the model's accuracy and training time. Table 7 shows the commonly used methods for searching the best hyperparameters.

Table 7. Hyperparameters for optimization.

Methods	Hyperparameters	Learning Rates	Number of Epochs	Batch Sizes	Iterations	Dropout Rates
Metaheuristics	PSO/IPSO/SPSO/SAPSO/PPSO	[7,21,49,70,72,94,102]	[7,49,102]	[7,44,49,52,55,71,102]	[82,102]	
	Hyper-Opt		[26]	[26]		
	SCA	[65]				
	IGOA	[37]				
	SOA/MTSO	[43]	[33]	[33]	[43]	
	GWO	[67]				[66]
	BER	[76]				
	WOA/IWOA	[35,63,64]	[63]	[35]	[35,38,64]	[63]
	FWA	[75]		[75]	[75]	
	IBFO	[46]			[46]	
	ABC	[62]				
	GA					[17]
	ACO	[6]		[6]		[6]
	SSA			[74]	[73,74]	
Others	Grid search	[30]		[30]	[30]	[34]
	Bayesian	[42,59,89,96,101]	[2,43,53,61,101]	[29,53,61,101]	[42]	[61,77]
	Unavailable	[1,8,23,27,28,84,85,92]	[1,8,28,78,84,85]	[1,23,28,79]	[1,18,27]	[42]

Note: PSO = particle swarm optimization; IPSO = improved particle swarm optimization; SPSO = selective particle swarm optimization; SAPSO = simulated annealing particle swarm optimization; PPSO = PCA-based particle swarm optimization; SCA = the sine cosine algorithm; IGOA = improved grasshopper optimization algorithm; SOA = seeker optimization algorithm; MTSO = modified tuna swarm optimization; GWO = gray wolf optimizer; BER = Al-Biruni Earth radius; WOA = whale optimization algorithm; IWOA = improved whale optimization algorithm; FWA = fireworks algorithm; IBFO = improved bacterial foraging optimization; ABC = artificial bee colony; GA = genetic algorithm; ACO = ant colony optimization; SSA = sparrow search algorithm.

Figure 4 divides the algorithms for finding the optimal hyperparameters into four types. Among the studies analyzed in this research, 31 were found to use metaheuristic algorithms; another 12 papers indicated the hyperparameters that were used but did not specify which method they used to search for the best hyperparameters.

How to determine the initial search values of the hyperparameters is debatable. Most studies apply different learning rate values in the Adam optimizer, in spite of the default value in most of the popular platforms being 0.001. As a result, how to determine the initial learning rate is a critical task. Smith et al. [106] proposed using a learning rate range test to find the best learning rate. In the beginning, the minimal value of the initial rate was set. After each iteration was performed, the learning rate continuously changed. When the best model performance was achieved, the searching task could stop, and the best learning rate could be generated.

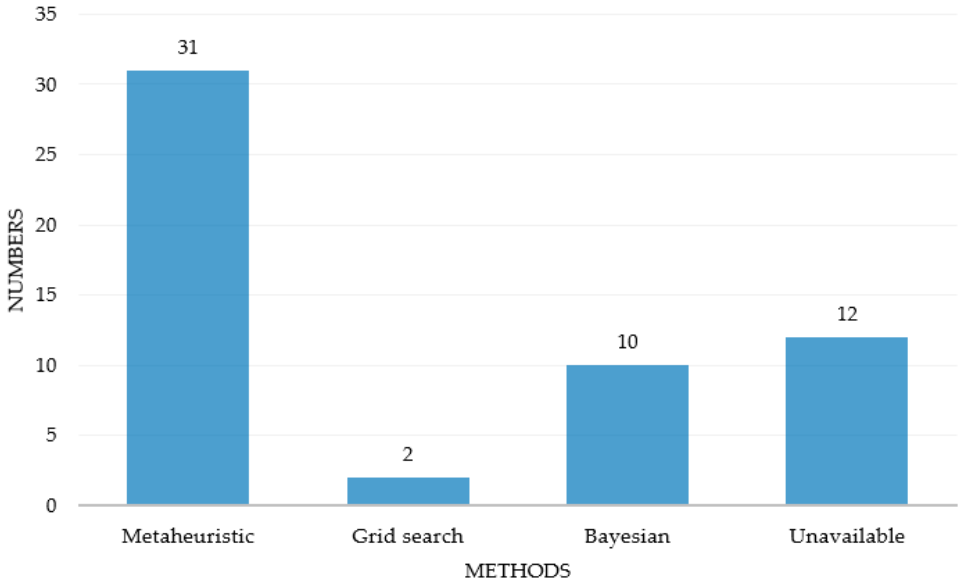

Figure 4. Methods of optimization.

Parameter adjustments aim to achieve the best accuracy and reduce the time and cost of model training. As shown in Figure 5, this study statistically analyzed hyperparameter optimization, and the results were consistent with the conclusion of Wang et al. [44], who found that the learning rate and batch size were the two most essential hyperparameters.

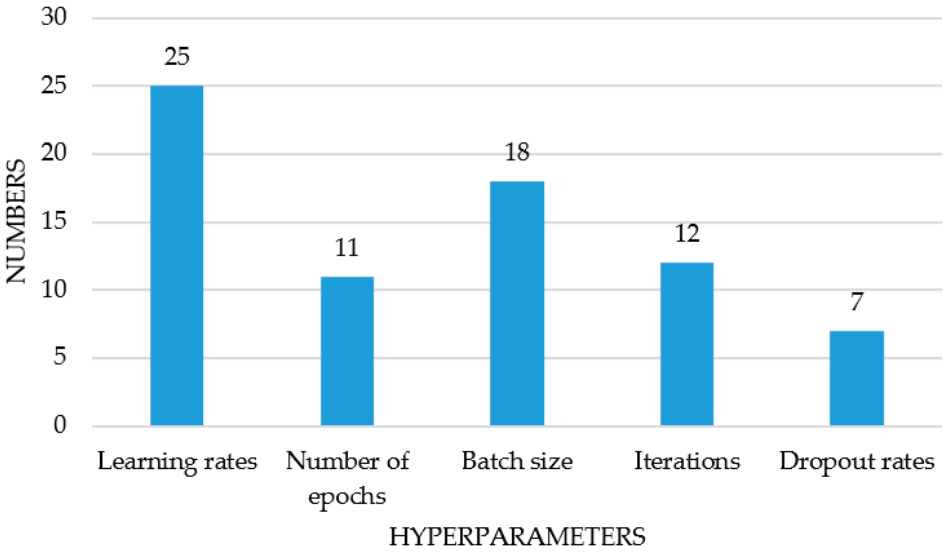

Figure 5. Optimized hyperparameters.

3. The Systemic and Global Optimization of DNNR Models

When training the DNNR model, the network architectures, optimizers, and hyperparameters will affect the prediction accuracy. However, it is unrealistic to consider all hyperparameters in the modeling process, because it will take too much model training time. Concurrently optimizing four sections could provide suitable DNNR models with higher regression accuracy. Table 8 lists the sections of optimization in terms of the architectures, optimizers, hyperparameters, and data preprocessing. It can be observed that only 1 out of 74 pieces of literature conducted four sections of optimization of DNNR models [31]. Furthermore, according to a number of studies [9,17,26,40,56,60,61,76,78,81,84,98–100], Figure 6 presents an overall optimization procedure for DNNR models.

Table 8. Sections of optimization.

Papers	Architectures	Optimizers	Hyperparameters	Data Preprocessing
[33]	X	X	X	X
[17,30,34,35,37,38,42,43,46,53]	X		X	X
[2,59,61–67]	X		X	
[48,51]		X		X
[6,21,26,29,44,49,52,55]			X	X
[57,58,60,68]	X			
[84–86]		X		
[7,70–77,82,89,94,96,101,102]			X	
[1,8,16,18–20,22–25,27,28,31,32,36,39–41,45,47,50,54]				X

Note: X = The paper contains the optimization section.

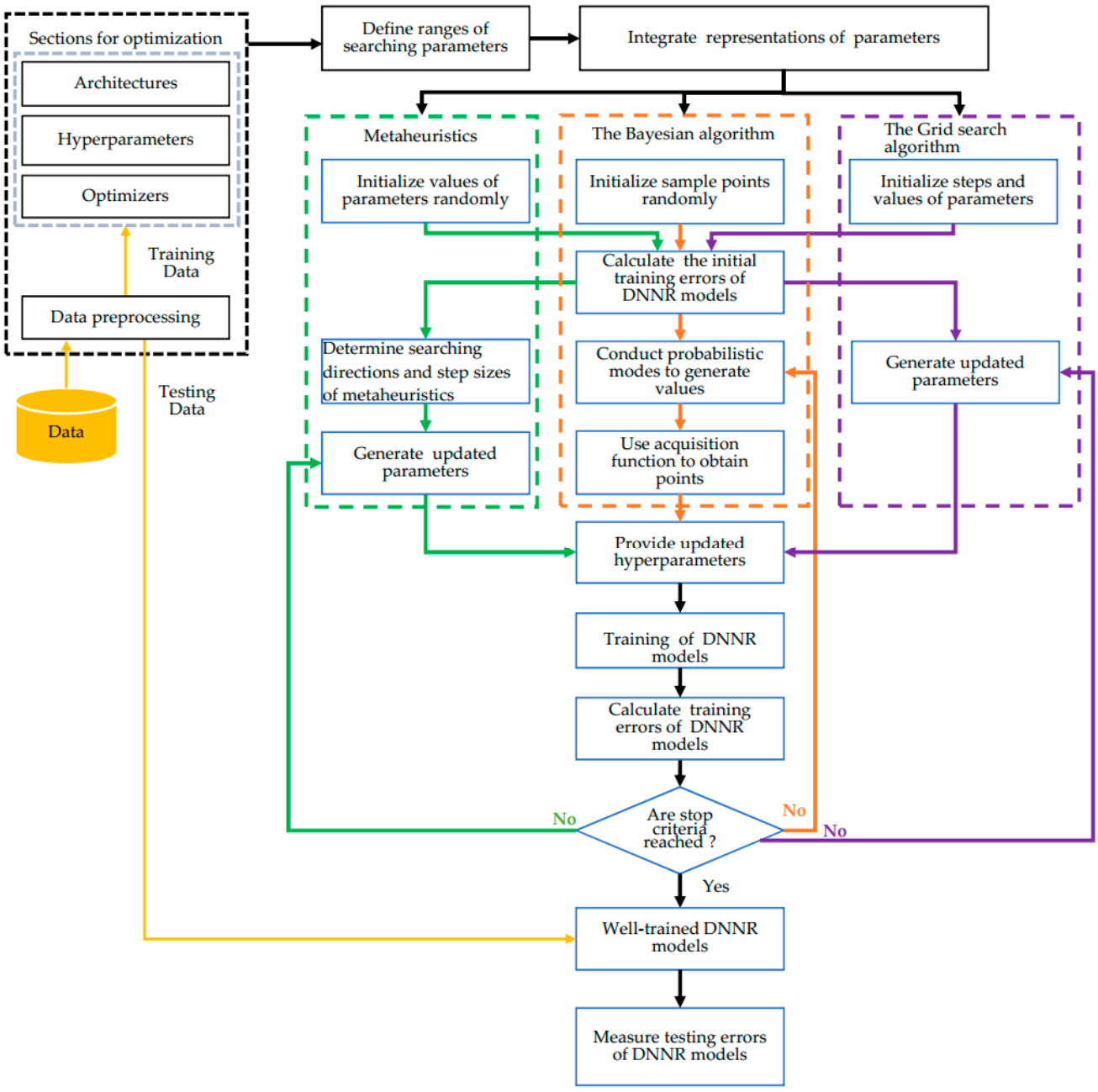

Figure 6. The optimization of DNNR models.

Data preprocessing is an essential step before data can be employed to model a DNNR. The selection of the data preprocessing procedures is mostly dependent on the data type and is related to the succeeding three sections of optimization.

Before optimizing the section parameters of DNNR models, the search ranges have to be defined. Then, due to the different types of section parameters, representations of the parameters must be integrated. The parameters in the presented optimization of a DNNR model can be represented by three types: real numbers, integers, and categories, as depicted in Table 9. The variables of the architectures, such as the number of hidden layers and nodes for each layer, are expressed in integers. Some hyperparameters, such as the number of epochs, batch sizes, and iterations, are in a form of integers. The categorical types refer to the types of optimizers and data preprocessing. The form of real numbers includes hyperparameters, such as learning rates and dropout rates. Figure 6 presents three methods used by DNNR models for optimization: metaheuristics, the Bayesian algorithm, and the grid search algorithm. The three algorithms operate according to similar logic. First, initial values are assigned randomly according to the ranges of the parameters, and new parameter values are calculated for each iteration. Before the stop criteria are reached, they are updated and tentative parameters are generated. Then, a training produce is performed for the DNNR model, and training errors are generated. When the stop criteria are reached, the finalized parameters for the DNNR modes are provided. Then, the training of the DNNR model is conducted to generate well-trained DNNR models. Finally, testing data are employed to measure the performance of the DNNR models. Two major issues related to the trade-off between the computational burden and the optimization of DNNR models (the number of parameters for optimization in sections and the search ranges of the parameters) are highlighted in Figure 6. The use of more parameters and larger search ranges can lead to better performance of the DNNR model; however, the computation time could increase unacceptably.

Table 9. Variable types for optimization.

Sections of Optimization	Types of Variables
Architectures	Integers
Optimizers	Categories
Hyperparameters	Integers, real numbers
Data preprocessing	Categories

4. Conclusions

It has been pointed out that the optimization of deep neural networks plays a crucial role in improving their performance [42,45,49,59,64,88,102,103]. This study investigated the preprocessing, network architectures, optimizers, and hyperparameters of deep neural networks for regression. The interactions among sections of optimization can lead to different regression accuracies and greatly influence forecasting performance. Thus, the aim of this investigation was to explore the optimization of DNNR models from the aspects of preprocessing, network architectures, optimizers, and hyperparameters. Furthermore, this study investigated the optimization of various sections both individually and simultaneously.

Studies [10,11] pointed out that optimizing the DNNR model is a non-deterministic polynomial-time hardness problem. The trade-off between computational burden and optimization has always been a crucial and challenging issue in developing deep learning models [107,108], especially optimizing four sections simultaneously in this study. The appropriate filtering strategy could be provided to balance the computation burden and optimization performance. First, proper values for parameters could be obtained and set from the literature. For example, some data preprocessing procedures are suitable for certain types of data patterns. Additionally, three hidden layers are enough for most problems of DNNR architectures [83]. Adam and RMSprop are often employed when selecting optimizers [51]. The value of 0.001 is recommended to be the initial learning rate [8,62]. The second strategy is to limit the appropriate searching ranges for parameters

from past studies. For example, the number of hidden nodes are mostly between 1 and 100 [33]. The range of the learning rate is set to be smaller than one and larger than zero [21]. However, due to the increased computational burden when performing all optimization tasks at the same time, proper filtered sections of optimization to strike a balance between computation efforts and global optimization are possibly another method to motivate future study.

The findings of this study can be illustrated as follows. First, the selection of data preprocessing procedures mostly depends on the data type, but appropriate preprocessing methods can generate more accurate regression results. Second, metaheuristics are the most popular and commonly used method for the selection of architectures and the tuning of hyperparameters. Third, the TensorFlow platform and the Adam or Adamx optimizer are usually employed when utilizing DNNR models. Finally, the integration of variable types is essential for optimizing DNNR models. Furthermore, although optimizing four sections simultaneously is a challenging task with a trade-off of computation effort, it is a critical issue to improve the overall regression performance of DNNR models. Some probable future research directions of the optimization of deep neural networks for regression are described as follows. To obtain accurate regression results in an acceptable training time, the setting of proper initial values and searching ranges for simultaneously optimizing four sections of DNNR is a direction worthy of further exploration.

Author Contributions: Conceptualization, P.-F.P.; data curation; C.-H.C. and J.-P.L.; formal analysis, C.-H.C., J.-P.L., Y.-M.C. and P.-F.P.; methodology, C.-H.C., J.-P.L. and P.-F.P.; visualization, C.-H.C. and P.-F.P.; writing—original draft, C.-H.C., J.-P.L., C.-J.L., C.-H.C. and P.-F.P.; writing—review and editing, C.-J.L. and P.-F.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ghimire, S.; Deo, R.C.; Raj, N.; Mi, J. Deep Learning Neural Networks Trained with MODIS Satellite-Derived Predictors for Long-Term Global Solar Radiation Prediction. *Energies* **2019**, *12*, 2407. [CrossRef]
2. Nasser, A.A.; Rashad, M.Z.; Hussein, S.E. A Two-Layer Water Demand Prediction System in Urban Areas Based on Micro-Services and LSTM Neural Networks. *IEEE Access* **2020**, *8*, 147647–147661. [CrossRef]
3. Hoekendijk, J.P.A.; Kellenberger, B.; Aarts, G.; Brasseur, S.; Poiesz, S.S.H.; Tuia, D. Counting Using Deep Learning Regression Gives Value to Ecological Surveys. *Sci. Rep.* **2021**, *11*, 23209. [CrossRef] [PubMed]
4. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep Learning in Agriculture: A Survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
5. Wang, H.; Lei, Z.; Zhang, X.; Zhou, B.; Peng, J. A Review of Deep Learning for Renewable Energy Forecasting. *Energy Convers. Manag.* **2019**, *198*, 111799. [CrossRef]
6. Xiong, Y.; Zhou, Y.; Wang, F.; Wang, S.; Wang, J.; Ji, J.; Wang, Z. Landslide Susceptibility Mapping Using Ant Colony Optimization Strategy and Deep Belief Network in Jiuzhaigou Region. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 11042–11057. [CrossRef]
7. Liu, J.; Jiang, R.; Zhu, D.; Zhao, J. Short-Term Subway Inbound Passenger Flow Prediction Based on AFC Data and PSO-LSTM Optimized Model. *Urban. Rail Transit.* **2022**, *8*, 56–66. [CrossRef]
8. Gao, Y.; Wang, R.; Zhou, E. Stock Prediction Based on Optimized LSTM and GRU Models. *Sci. Program.* **2021**, *2021*, 4055281. [CrossRef]
9. Shrestha, A.; Mahmood, A. Review of Deep Learning Algorithms and Architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
10. Dong, S.; Wang, P.; Abbas, K. A Survey on Deep Learning and Its Applications. *Comput. Sci. Rev.* **2021**, *40*, 100379. [CrossRef]
11. Abd Elaziz, M.; Dahou, A.; Abualigah, L.; Yu, L.; Alshinwan, M.; Khasawneh, A.M.; Lu, S. Advanced Metaheuristic Optimization Techniques in Applications of Deep Neural Networks: A Review. *Neural Comput. Appl.* **2021**, *33*, 14079–14099. [CrossRef]
12. Akay, B.; Karaboga, D.; Akay, R. A Comprehensive Survey on Optimizing Deep Learning Models by Metaheuristics. *Artif. Intell. Rev.* **2022**, *55*, 829–894. [CrossRef]
13. Zhan, Z.-H.; Li, J.-Y.; Zhang, J. Evolutionary Deep Learning: A Survey. *Neurocomputing* **2022**, *483*, 42–58. [CrossRef]
14. Shickel, B.; Tighe, P.J.; Bihorac, A.; Rashidi, P. Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis. *IEEE J. Biomed. Health Inform.* **2017**, *22*, 1589–1604. [CrossRef] [PubMed]
15. Abram, K.J.; McCloskey, D. A Comprehensive Evaluation of Metabolomics Data Preprocessing Methods for Deep Learning. *Metabolites* **2022**, *12*, 202. [CrossRef]

16. Han, D.; Yang, X.; Li, G.; Wang, S.; Wang, Z.; Zhao, J. Highway Traffic Speed Prediction in Rainy Environment Based on APSO-GRU. *J. Adv. Transp.* **2021**, *2021*, 4060740. [CrossRef]
17. Tsokov, S.; Lazarova, M.; Aleksieva-Petrova, A. A Hybrid Spatiotemporal Deep Model Based on CNN and LSTM for Air Pollution Prediction. *Sustainability* **2022**, *14*, 5104. [CrossRef]
18. Jia, P.; Liu, H.; Wang, S.; Wang, P. Research on a Mine Gas Concentration Forecasting Model Based on a GRU Network. *IEEE Access* **2020**, *8*, 38023–38031. [CrossRef]
19. Shao, B.; Song, D.; Bian, G.; Zhao, Y. A Hybrid Approach by CEEMDAN-Improved PSO-LSTM Model for Network Traffic Prediction. *Secur. Commun. Netw.* **2022**, *2022*, 4975288. [CrossRef]
20. Yan, J.; Gao, Y.; Yu, Y.; Xu, H.; Xu, Z. A Prediction Model Based on Deep Belief Network and Least Squares SVR Applied to Cross-Section Water Quality. *Water* **2020**, *12*, 1929. [CrossRef]
21. Zhang, Y. Short-Term Power Load Forecasting Based on SAPSO-CNN-LSTM Model Considering Autocorrelated Errors. *Math. Probl. Eng.* **2022**, *2022*, 2871889. [CrossRef]
22. Abidi, M.H.; Alkhalefah, H.; Mohammed, M.K.; Umer, U.; Qudeiri, J.E.A. Optimal Scheduling of Flexible Manufacturing System Using Improved Lion-Based Hybrid Machine Learning Approach. *IEEE Access* **2020**, *8*, 96088–96114. [CrossRef]
23. Chen, X. Emotional Calculation Method of Rural Tourist Based on Improved SPCA-LSTM Algorithm. *J. Sens.* **2022**, *2022*, 3365498. [CrossRef]
24. Cheng, T.; Harrou, F.; Kadri, F.; Sun, Y.; Leiknes, T. Forecasting of Wastewater Treatment Plant Key Features Using Deep Learning-Based Models: A Case Study. *IEEE Access* **2020**, *8*, 184475–184485. [CrossRef]
25. Zhao, A.; Mi, L.; Xue, X.; Xi, J.; Jiao, Y. Heating Load Prediction of Residential District Using Hybrid Model Based on CNN. *Energy Build.* **2022**, *266*, 112122. [CrossRef]
26. Ghimire, S.; Bhandari, B.; Casillas-Pérez, D.; Deo, R.C.; Salcedo-Sanz, S. Hybrid Deep CNN-SVR Algorithm for Solar Radiation Prediction Problems in Queensland, Australia. *Eng. Appl. Artif. Intell.* **2022**, *112*, 104860. [CrossRef]
27. Liu, W.; Yu, H.; Yang, L.; Yin, Z.; Zhu, M.; Wen, X. Deep Learning-Based Predictive Framework for Groundwater Level Forecast in Arid Irrigated Areas. *Water* **2021**, *13*, 2558. [CrossRef]
28. Wang, F.; Ma, S.; Wang, H.; Li, Y.; Zhang, J. Prediction of NO X Emission for Coal-Fired Boilers Based on Deep Belief Network. *Control Eng. Pract.* **2018**, *80*, 26–35. [CrossRef]
29. Lian, P.; Liu, H.; Wang, X.; Guo, R. Soft Sensor Based on DBN-IPSO-SVR Approach for Rotor Thermal Deformation Prediction of Rotary Air-Preheater. *Measurement* **2020**, *165*, 108109. [CrossRef]
30. Song, J.; Xue, G.; Ma, Y.; Li, H.; Pan, Y.; Hao, Z. An Indoor Temperature Prediction Framework Based on Hierarchical Attention Gated Recurrent Unit Model for Energy Efficient Buildings. *IEEE Access* **2019**, *7*, 157268–157283. [CrossRef]
31. He, Q.-Q.; Wu, C.; Si, Y.-W. LSTM with Particle Swam Optimization for Sales Forecasting. *Electron. Commer. Res. Appl.* **2022**, *51*, 101118. [CrossRef]
32. Li, X.; Liu, B.; Qian, W.; Rao, G.; Chen, L.; Cui, J. Design of Soft-Sensing Model for Alumina Concentration Based on Improved Deep Belief Network. *Processes* **2022**, *10*, 2537. [CrossRef]
33. Jiang, H.; Hu, W.; Xiao, L.; Dong, Y. A Decomposition Ensemble Based Deep Learning Approach for Crude Oil Price Forecasting. *Resour. Policy* **2022**, *78*, 102855. [CrossRef]
34. Gao, B.; Huang, X.; Shi, J.; Tai, Y.; Zhang, J. Hourly Forecasting of Solar Irradiance Based on CEEMDAN and Multi-Strategy CNN-LSTM Neural Networks. *Renew. Energy* **2020**, *162*, 1665–1683. [CrossRef]
35. Yang, H.; Liu, S. Water Quality Prediction in Sea Cucumber Farming Based on a GRU Neural Network Optimized by an Improved Whale Optimization Algorithm. *PeerJ Comput. Sci.* **2022**, *8*, e1000. [CrossRef] [PubMed]
36. Zhang, Y.; Tang, J.; Cheng, Y.; Huang, L.; Guo, F.; Yin, X.; Li, N. Prediction of Landslide Displacement with Dynamic Features Using Intelligent Approaches. *Int. J. Min. Sci. Technol.* **2022**, *32*, 539–549. [CrossRef]
37. Hu, H.; Xia, X.; Luo, Y.; Zhang, C.; Nazir, M.S.; Peng, T. Development and Application of an Evolutionary Deep Learning Framework of LSTM Based on Improved Grasshopper Optimization Algorithm for Short-Term Load Forecasting. *J. Build. Eng.* **2022**, *57*, 104975. [CrossRef]
38. Wang, H.; Xiong, M.; Chen, H.; Liu, S. Multi-Step Ahead Wind Speed Prediction Based on a Two-Step Decomposition Technique and Prediction Model Parameter Optimization. *Energy Rep.* **2022**, *8*, 6086–6100. [CrossRef]
39. Duan, J.; Chang, M.; Chen, X.; Wang, W.; Zuo, H.; Bai, Y.; Chen, B. A Combined Short-Term Wind Speed Forecasting Model Based on CNN–RNN and Linear Regression Optimization Considering Error. *Renew. Energy* **2022**, *200*, 788–808. [CrossRef]
40. Liu, H.; Mi, X.; Li, Y.; Duan, Z.; Xu, Y. Smart Wind Speed Deep Learning Based Multi-Step Forecasting Model Using Singular Spectrum Analysis, Convolutional Gated Recurrent Unit Network and Support Vector Regression. *Renew. Energy* **2019**, *143*, 842–854. [CrossRef]
41. Shang, Z.; Wen, Q.; Chen, Y.; Zhou, B.; Xu, M. Wind Speed Forecasting Using Attention-Based Causal Convolutional Network and Wind Energy Conversion. *Energies* **2022**, *15*, 2881. [CrossRef]
42. Wang, S.; Qin, C.; Feng, Q.; Javadpour, F.; Rui, Z. A Framework for Predicting the Production Performance of Unconventional Resources Using Deep Learning. *Appl. Energy* **2021**, *295*, 117016. [CrossRef]
43. Tuerxun, W.; Xu, C.; Guo, H.; Guo, L.; Zeng, N.; Cheng, Z. An Ultra-short-term Wind Speed Prediction Model Using LSTM Based on Modified Tuna Swarm Optimization and Successive Variational Mode Decomposition. *Energy Sci. Eng.* **2022**, *10*, 3001–3022. [CrossRef]

44. Wang, Y.; Zhang, W.; Sun, J.; Wang, L.; Song, X.; Zhao, X. Survival Prediction Model for Patients with Esophageal Squamous Cell Carcinoma Based on the Parameter-Optimized Deep Belief Network Using the Improved Archimedes Optimization Algorithm. *Comput. Math. Methods Med.* **2022**, *2022*, 1924906. [CrossRef] [PubMed]
45. Mohammed, G.P.; Alasmari, N.; Alsolai, H.; Alotaibi, S.S.; Alotaibi, N.; Mohsen, H. Autonomous Short-Term Traffic Flow Prediction Using Pelican Optimization with Hybrid Deep Belief Network in Smart Cities. *Appl. Sci.* **2022**, *12*, 10828. [CrossRef]
46. Zhang, Y.; Gao, G. Optimization and Evaluation of an Intelligent Short-Term Blood Glucose Prediction Model Based on Noninvasive Monitoring and Deep Learning Techniques. *J. Healthc. Eng.* **2022**, *2022*, 8956850. [CrossRef]
47. Yang, C.-H.; Chen, B.-H.; Wu, C.-H.; Chen, K.-C.; Chuang, L.-Y. Deep Learning for Forecasting Electricity Demand in Taiwan. *Mathematics* **2022**, *10*, 2547. [CrossRef]
48. Gao, S.; Huang, Y.; Zhang, S.; Han, J.; Wang, G.; Zhang, M.; Lin, Q. Short-Term Runoff Prediction with GRU and LSTM Networks without Requiring Time Step Optimization during Sample Generation. *J. Hydrol.* **2020**, *589*, 125188. [CrossRef]
49. Li, J.; Zhang, Z.; Wang, X.; Yan, W. Intelligent Decision-Making Model in Preventive Maintenance of Asphalt Pavement Based on PSO-GRU Neural Network. *Adv. Eng. Inform.* **2022**, *51*, 101525. [CrossRef]
50. Meng, X.; Wang, R.; Zhang, X.; Wang, M.; Ma, H.; Wang, Z. Hybrid Neural Network Based on GRU with Uncertain Factors for Forecasting Ultra-Short-Term Wind Power. In Proceedings of the 2020 2nd International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 23–25 October 2020; pp. 1–6.
51. Saini, V.K.; Bhardwaj, B.; Gupta, V.; Kumar, R.; Mathur, A. Gated recurrent unit (gru) based short term forecasting for wind energy estimation. In Proceedings of the 2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 10–11 December 2020; pp. 1–6.
52. Li, Q.; Chai, X.; Zhang, C.; Wang, X.; Ma, W. Prediction Model of Ischemic Stroke Recurrence Using PSO-LSTM in Mobile Medical Monitoring System. *Comput. Intell. Neurosci.* **2022**, *2022*, 8936103. [CrossRef]
53. Qiu, K.; Li, J.; Chen, D. Optimized Long Short-Term Memory (LSTM) Network for Performance Prediction in Unconventional Reservoirs. *Energy Rep.* **2022**, *8*, 15436–15445. [CrossRef]
54. Zhou, B.; Ma, X.; Luo, Y.; Yang, D. Wind Power Prediction Based on LSTM Networks and Nonparametric Kernel Density Estimation. *IEEE Access* **2019**, *7*, 165279–165292. [CrossRef]
55. Chen, F.; Gao, X.; Xia, X.; Xu, J. Using LSTM and PSO Techniques for Predicting Moisture Content of Poplar Fibers by Impulse-Cyclone Drying. *PLoS ONE* **2022**, *17*, e0266186. [CrossRef] [PubMed]
56. Aladag, C.H. Architecture Selection in Neural Networks by Statistical and Machine Learning. *Orient. J. Comp. Sci. Technol.* **2019**, *12*, 76–89. [CrossRef]
57. Sen, J.; Mehtab, S. Accurate Stock Price Forecasting Using Robust and Optimized Deep Learning Models. In Proceedings of the 2021 International Conference on Intelligent Technologies (CONIT), Hubli, India, 25–27 June 2021; pp. 1–9.
58. Shi, X.; Huang, G.; Hao, X.; Yang, Y.; Li, Z. A Synchronous Prediction Model Based on Multi-Channel CNN with Moving Window for Coal and Electricity Consumption in Cement Calcination Process. *Sensors* **2021**, *21*, 4284. [CrossRef] [PubMed]
59. Itakura, K.; Saito, Y.; Suzuki, T.; Kondo, N.; Hosoi, F. Estimation of Citrus Maturity with Florescence Spectroscopy Using Deep Learning. *Horticulturae* **2018**, *5*, 2. [CrossRef]
60. Kulshrestha, A.; Krishnaswamy, V.; Sharma, M. Bayesian BILSTM Approach for Tourism Demand Forecasting. *Ann. Tour. Res.* **2020**, *83*, 102925. [CrossRef]
61. Di, Y.; Gao, M.; Feng, F.; Li, Q.; Zhang, H. A New Framework for Winter Wheat Yield Prediction Integrating Deep Learning and Bayesian Optimization. *Agronomy* **2022**, *12*, 3194. [CrossRef]
62. Yalçın, S.; Panchal, S.; Herdem, M.S. A CNN-ABC Model for Estimation and Optimization of Heat Generation Rate and Voltage Distributions of Lithium-Ion Batteries for Electric Vehicles. *Int. J. Heat. Mass. Transf.* **2022**, *199*, 123486. [CrossRef]
63. Rajamoorthy, R.; Saraswathi, H.V.; Devaraj, J.; Kasinathan, P.; Elavarasan, R.M.; Arunachalam, G.; Mostafa, T.M.; Mihet-Popa, L. A Hybrid Sailfish Whale Optimization and Deep Long Short-Term Memory (SWO-DLSTM) Model for Energy Efficient Autonomy in India by 2048. *Sustainability* **2022**, *14*, 1355. [CrossRef]
64. Ge, S.; Gao, W.; Cui, S.; Chen, X.; Wang, S. Safety Prediction of Shield Tunnel Construction Using Deep Belief Network and Whale Optimization Algorithm. *Autom. Constr.* **2022**, *142*, 104488. [CrossRef]
65. Sun, L.; Qin, H.; Przystupa, K.; Majka, M.; Kochan, O. Individualized Short-Term Electric Load Forecasting Using Data-Driven Meta-Heuristic Method Based on LSTM Network. *Sensors* **2022**, *22*, 7900. [CrossRef] [PubMed]
66. Hakim, W.L.; Rezaie, F.; Nur, A.S.; Panahi, M.; Khosravi, K.; Lee, C.-W.; Lee, S. Convolutional Neural Network (CNN) with Metaheuristic Optimization Algorithms for Landslide Susceptibility Mapping in Icheon, South Korea. *J. Environ. Manag.* **2022**, *305*, 114367. [CrossRef]
67. Pan, J.; Jing, B.; Jiao, X.; Wang, S. Analysis and Application of Grey Wolf Optimizer-Long Short-Term Memory. *IEEE Access* **2020**, *8*, 121460–121468. [CrossRef]
68. Mahdaddi, A.; Meshoul, S.; Belguidoum, M. EA-Based Hyperparameter Optimization of Hybrid Deep Learning Models for Effective Drug-Target Interactions Prediction. *Expert Syst. Appl.* **2021**, *185*, 115525. [CrossRef]
69. Wang, Y.; Wei, S.; Yang, W.; Chai, Y.; Li, P. Construction of Offline Predictive Controller for Wind Farm Based on CNN–GRNN. *Control Eng. Pract.* **2022**, *127*, 105290. [CrossRef]
70. 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]

71. Hu, Y.; Wei, R.; Yang, Y.; Li, X.; Huang, Z.; Liu, Y.; He, C.; Lu, H. Performance Degradation Prediction Using LSTM with Optimized Parameters. *Sensors* **2022**, *22*, 2407. [CrossRef]
72. Gao, J.; Wang, X.; Yang, W. SPSO-DBN Based Compensation Algorithm for Lackness of Electric Energy Metering in Micro-Grid. *Alex. Eng. J.* **2022**, *61*, 4585–4594. [CrossRef]
73. Gao, S.; Xu, L.; Zhang, Y.; Pei, Z. Rolling Bearing Fault Diagnosis Based on SSA Optimized Self-Adaptive DBN. *ISA Trans.* **2022**, *128*, 485–502. [CrossRef]
74. Wang, X.; Yan, C.; Liu, W.; Liu, X. Research on Carbon Emissions Prediction Model of Thermal Power Plant Based on SSA-LSTM Algorithm with Boiler Feed Water Influencing Factors. *Sustainability* **2022**, *14*, 15988. [CrossRef]
75. Shao, B.; Song, D.; Bian, G.; Zhao, Y. Wind Speed Forecast Based on the LSTM Neural Network Optimized by the Firework Algorithm. *Adv. Mater. Sci. Eng.* **2021**, *2021*, 4874757. [CrossRef]
76. Eid, M.M.; El-Kenawy, E.-S.M.; Khodadadi, N.; Mirjalili, S.; Khodadadi, E.; Abotaleb, M.; Alharbi, A.H.; Abdelhamid, A.A.; Ibrahim, A.; Amer, G.M.; et al. Meta-Heuristic Optimization of LSTM-Based Deep Network for Boosting the Prediction of Monkeypox Cases. *Mathematics* **2022**, *10*, 3845. [CrossRef]
77. Yin, X.; Liu, Q.; Huang, X.; Pan, Y. Real-Time Prediction of Rockburst Intensity Using an Integrated CNN-Adam-BO Algorithm Based on Microseismic Data and Its Engineering Application. *Tunn. Undergr. Space Technol.* **2021**, *117*, 104133. [CrossRef]
78. Yu, Z.; Sun, Y.; Zhang, J.; Zhang, Y.; Liu, Z. Gated Recurrent Unit Neural Network (GRU) Based on Quantile Regression (QR) Predicts Reservoir Parameters through Well Logging Data. *Front. Earth Sci.* **2023**, *11*, 1087385. [CrossRef]
79. Wei, D.; Wang, J.; Niu, X.; Li, Z. Wind Speed Forecasting System Based on Gated Recurrent Units and Convolutional Spiking Neural Networks. *Appl. Energy* **2021**, *292*, 116842. [CrossRef]
80. Wang, J.; Li, Q.; Zhang, H.; Wang, Y. A Deep-Learning Wind Speed Interval Forecasting Architecture Based on Modified Scaling Approach with Feature Ranking and Two-Output Gated Recurrent Unit. *Expert Syst. Appl.* **2023**, *211*, 118419. [CrossRef]
81. Li, Y.-Q.; Zhao, H.-W.; Yue, Z.-X.; Li, Y.-W.; Zhang, Y.; Zhao, D.-C. Real-Time Intelligent Prediction Method of Cable's Fundamental Frequency for Intelligent Maintenance of Cable-Stayed Bridges. *Sustainability* **2023**, *15*, 4086. [CrossRef]
82. Xu, Y.; Zhang, J.; Long, Z.; Tang, H.; Zhang, X. Hourly Urban Water Demand Forecasting Using the Continuous Deep Belief Echo State Network. *Water* **2019**, *11*, 351. [CrossRef]
83. Uzair, M.; Jamil, N. Effects of Hidden Layers on the Efficiency of Neural Networks. In Proceedings of the 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 5–7 November 2020; pp. 1–6.
84. Li, W.; Wu, H.; Zhu, N.; Jiang, Y.; Tan, J.; Guo, Y. Prediction of Dissolved Oxygen in a Fishery Pond Based on Gated Recurrent Unit (GRU). *Inf. Process. Agric.* **2021**, *8*, 185–193. [CrossRef]
85. Hu, Z. Prediction Model of Rotor Yarn Quality Based on CNN-LSTM. *J. Sens.* **2022**, *2022*, 3955047. [CrossRef]
86. Li, H.; Zhao, Z.; Du, X. Research and Application of Deformation Prediction Model for Deep Foundation Pit Based on LSTM. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9407999. [CrossRef]
87. Chen, J.; Zhang, H.; Zhang, W.; Du, X.; Zhang, Y.; Li, S. Correlated Regression Feature Learning for Automated Right Ventricle Segmentation. *IEEE J. Transl. Eng. Health Med.* **2018**, *6*, 1800610. [CrossRef]
88. Aslam, S.; Ayub, N.; Farooq, U.; Alvi, M.J.; Albogamy, F.R.; Rukh, G.; Haider, S.I.; Azar, A.T.; Bukhsh, R. Towards Electric Price and Load Forecasting Using CNN-Based Ensembler in Smart Grid. *Sustainability* **2021**, *13*, 12653. [CrossRef]
89. Zou, M.; Zhu, S.; Gu, J.; Korunovic, L.M.; Djokic, S.Z. Heating and Lighting Load Disaggregation Using Frequency Components and Convolutional Bidirectional Long Short-Term Memory Method. *Energies* **2021**, *14*, 4831. [CrossRef]
90. Miao, S.; Wang, Z.J.; Liao, R. A CNN Regression Approach for Real-Time 2D/3D Registration. *IEEE Trans. Med. Imaging* **2016**, *35*, 1352–1363. [CrossRef]
91. Zhao, Y.; Hu, H.; Song, C.; Wang, Z. Predicting Compressive Strength of Manufactured-Sand Concrete Using Conventional and Metaheuristic-Tuned Artificial Neural Network. *Measurement* **2022**, *194*, 110993. [CrossRef]
92. Rather, A.M. LSTM-Based Deep Learning Model for Stock Prediction and Predictive Optimization Model. *EURO J. Decis. Process.* **2021**, *9*, 100001. [CrossRef]
93. Pattana-Anake, V.; Joseph, F.J.J. Hyper Parameter Optimization of Stack LSTM Based Regression for PM 2.5 Data in Bangkok. In Proceedings of the 2022 7th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand, 19–20 May 2022; pp. 13–17.
94. Bian, J.; Wang, L.; Scherer, R.; Wozniak, M.; Zhang, P.; Wei, W. Abnormal Detection of Electricity Consumption of User Based on Particle Swarm Optimization and Long Short Term Memory With the Attention Mechanism. *IEEE Access* **2021**, *9*, 47252–47265. [CrossRef]
95. Yan, J.; Chen, X.; Yu, Y.; Zhang, X. Application of a Parallel Particle Swarm Optimization-Long Short Term Memory Model to Improve Water Quality Data. *Water* **2019**, *11*, 1317. [CrossRef]
96. Kaselimi, M.; Doulamis, N.; Doulamis, A.; Voulodimos, A.; Protopapadakis, E. Bayesian-Optimized Bidirectional LSTM Regression Model for Non-Intrusive Load Monitoring. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 2747–2751.
97. Wang, Y.; Feng, B.; Hua, Q.-S.; Sun, L. Short-Term Solar Power Forecasting: A Combined Long Short-Term Memory and Gaussian Process Regression Method. *Sustainability* **2021**, *13*, 3665. [CrossRef]
98. Islam, M.S.; Hossain, E. Foreign Exchange Currency Rate Prediction Using a GRU-LSTM Hybrid Network. *Soft Comput. Lett.* **2021**, *3*, 100009. [CrossRef]

99. Violos, J.; Tsanakas, S.; Theodoropoulos, T.; Leivadeas, A.; Tserpes, K.; Varvarigou, T. Hypertuning GRU Neural Networks for Edge Resource Usage Prediction. In Proceedings of the 2021 IEEE Symposium on Computers and Communications (ISCC), Athens, Greece, 5–8 September 2021; pp. 1–8.
100. Wang, J.; Cao, J.; Yuan, S.; Cheng, M. Short-Term Forecasting of Natural Gas Prices by Using a Novel Hybrid Method Based on a Combination of the CEEMDAN-SE-and the PSO-ALS-Optimized GRU Network. *Energy* **2021**, *233*, 121082. [CrossRef]
101. Cao, M.; Zhang, T.; Liu, Y.; Wang, Y.; Shi, Z. A Bayesian Optimization Hyperband-Optimized Incremental Deep Belief Network for Online Battery Behaviour Modelling for a Satellite Simulator. *J. Energy Storage* **2023**, *58*, 106348. [CrossRef]
102. Wang, S.; Lin, X.; Qi, X.; Li, H.; Yang, J. Landslide Susceptibility Analysis Based on a PSO-DBN Prediction Model in an Earthquake-Stricken Area. *Front. Environ. Sci.* **2022**, *10*, 912523. [CrossRef]
103. Cao, M.; Zhang, T.; Wang, J.; Liu, Y. A Deep Belief Network Approach to Remaining Capacity Estimation for Lithium-Ion Batteries Based on Charging Process Features. *J. Energy Storage* **2022**, *48*, 103825. [CrossRef]
104. Choi, R.Y.; Coyner, A.S.; Kalpathy-Cramer, J.; Chiang, M.F.; Campbell, J.P. Introduction to Machine Learning, Neural Networks, and Deep Learning. *Transl. Vis. Sci. Technol.* **2020**, *9*, 14.
105. Zhao, X.; Liu, D.; Yan, X. Diameter Prediction of Silicon Ingots in the Czochralski Process Based on a Hybrid Deep Learning Model. *Crystals* **2022**, *13*, 36. [CrossRef]
106. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 464–472.
107. Lee, S.; Kim, J.; Kang, H.; Kang, D.-Y.; Park, J. Genetic Algorithm Based Deep Learning Neural Network Structure and Hyperparameter Optimization. *Appl. Sci.* **2021**, *11*, 744. [CrossRef]
108. Shah, B.; Bhavsar, H. Time Complexity in Deep Learning Models. *Procedia Comput. Sci.* **2022**, *215*, 202–210. [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.